



Starsim: A flexible framework for agent-based modeling of health and disease

Cliff Kerr, Robyn Stuart, Romesh Abeysuriya, Jamie Cohen, Paula Sanz-Leon,
Keith Alexander Ashe, Luojun Yang, Alina Muellenmeister, Clinton Collins, Dan Klein

July 10, 2024

tl;dr version

- Starsim is an **open-access modeling framework** for making **agent-based models** of **multiple diseases**
- It's based on our previous disease-specific models for COVID, family planning, HPV, etc.
- It's designed to be easy to use: >200 people trained
- It's still in development, and we're actively seeking collaborators – get in touch with modeling questions!



Agent-based modeling training, Nairobi, April 2024

Code:

- Starsim: <https://starsim.org/>
- Covasim: <http://covasim.org>
- FPSim: <http://fpsim.org>
- HPVsim: <http://hpvsim.org>

Methods papers:

- Starsim: in prep!
- Covasim: Kerr et al 2021, [PLoS CB](#)
- FPSim: O'Brien et al 2023, [Nature WH](#)
- HPVsim: Stuart et al 2024, [PLoS CB](#)

Outline

1. **Motivation & genesis:** why / how did Starsim come about?
 - Origins in COVID-19
 - Expansion to other diseases/health areas
 - Evolution of an independent framework
2. **Design and implementation**
3. **Application:** Modeling co-circulating STIs
4. **Summary**

Motivation & genesis

Starsim's genesis

- Initially developed by the Institute of Disease Modeling, an embedded research group within the Gates Foundation
 - Provides internal support
 - Conducts scientific research
 - Develops **open-source modeling tools**
- In March 2020, urgent need for fast-paced modeling tools
 - IDM's existing models were not fast or nimble enough to adapt

News | Coronavirus pandemic

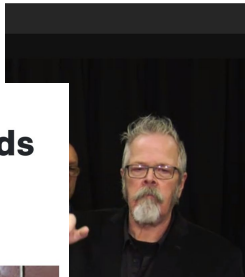
Kenya bans public events after first case of coronavirus

Sporting events, open-air religious meetings and 'all events of a huge public nature' have been suspended.



Kenya goes into dusk-to-dawn curfew as contact tracing expands to 1,000

SATURDAY MARCH 28 2020



Kenya's COVID-19 lockdown is forcing people to make difficult food and household energy decisions

Published: April 19, 2021 5:57pm SAST



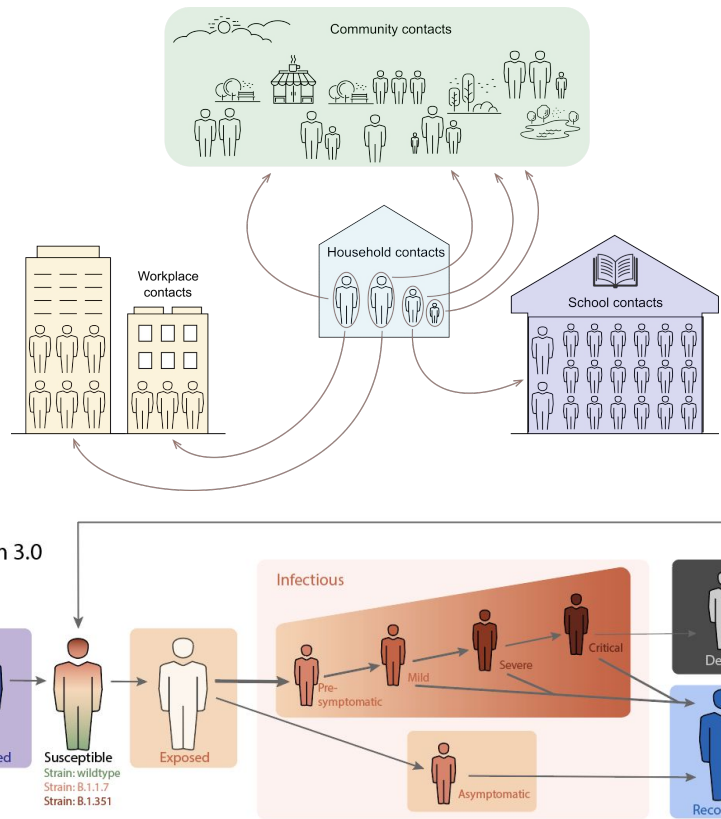
Modeling at pandemic pace required new tools

Covasim is an agent-based COVID-19 model purpose-built to explore complex scenarios

- **Clear purpose:** Simple design
- **Changing questions:** Flexible, easy to use
- **Fast development:** Pure python
- **Rigorous science:** calibration and local data

Enabled focus on the **science:**

- Infection natural history
- Detailed contact networks
- Individual-level heterogeneity
- Modeling antibody-based immunity
- Co-transmitting variants of concern



Many questions throughout the pandemic



How much testing & tracing do we need to do? How would this change if we opened up more?

Is it OK to open schools? How can we do it safely?

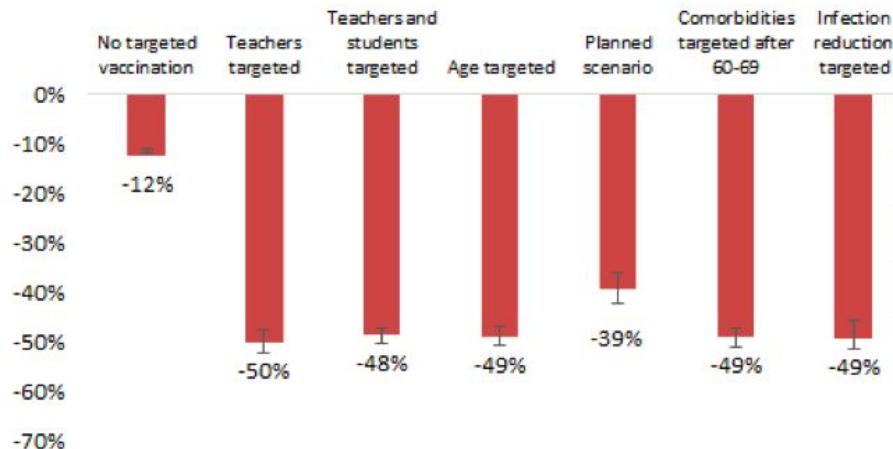
Is it OK to open borders? How can we do it safely?

What will be the impact of new variants?

What would be the impact of different vaccination roll-out strategies?

2020

Percentage of hospitalizations averted Jun-Dec 2021
8,400 vaccinations per day
Compared with no vaccination



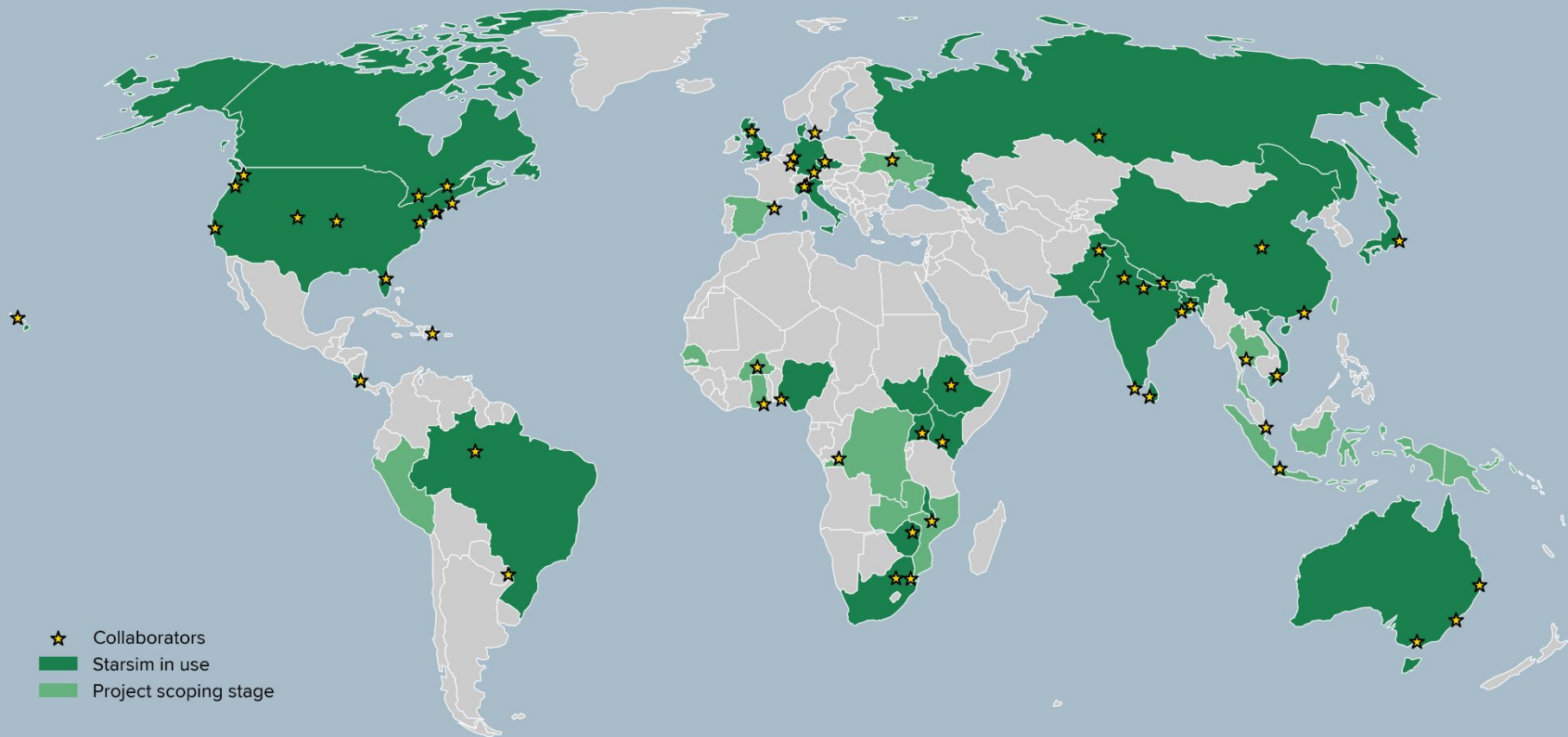
2021

Answers from WHO study: "We estimated vaccination coverage sufficient to ensure no school closures would be required to break the spread of symptomatic testing exceeded daily hospitalizations targeted to the most vulnerable groups have the greatest impact on the health system"

Pham QD et al, [https://doi.org/10.1016/S2214-109X\(21\)00103-0](https://doi.org/10.1016/S2214-109X(21)00103-0)

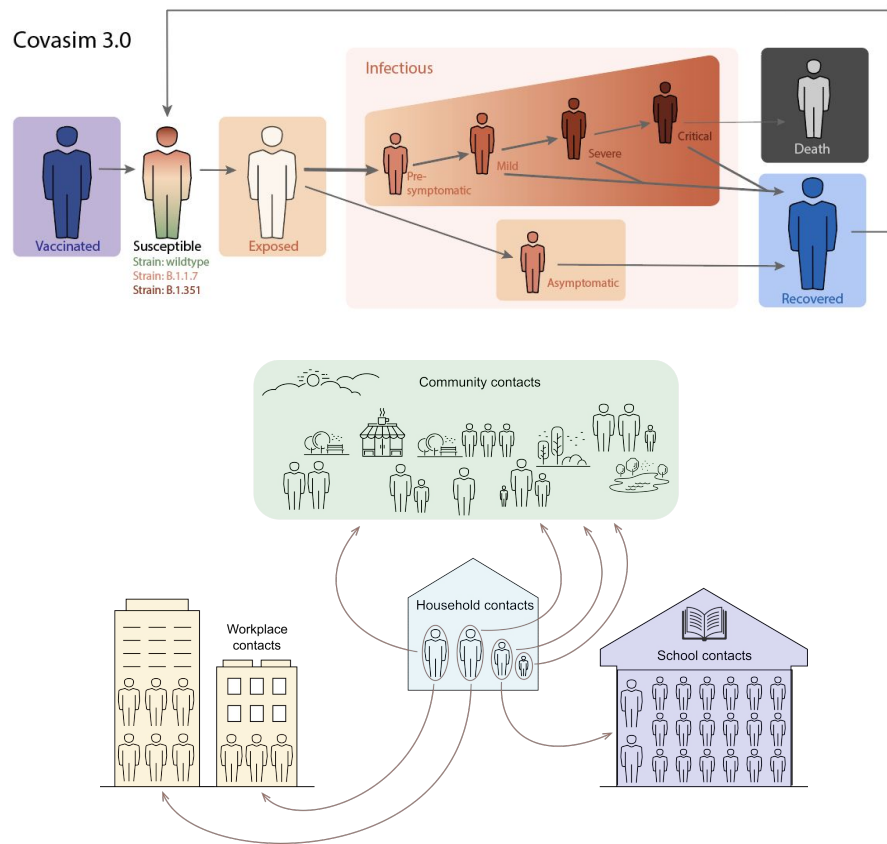
WHO report

Covasim quickly spread globally



Covasim's structure

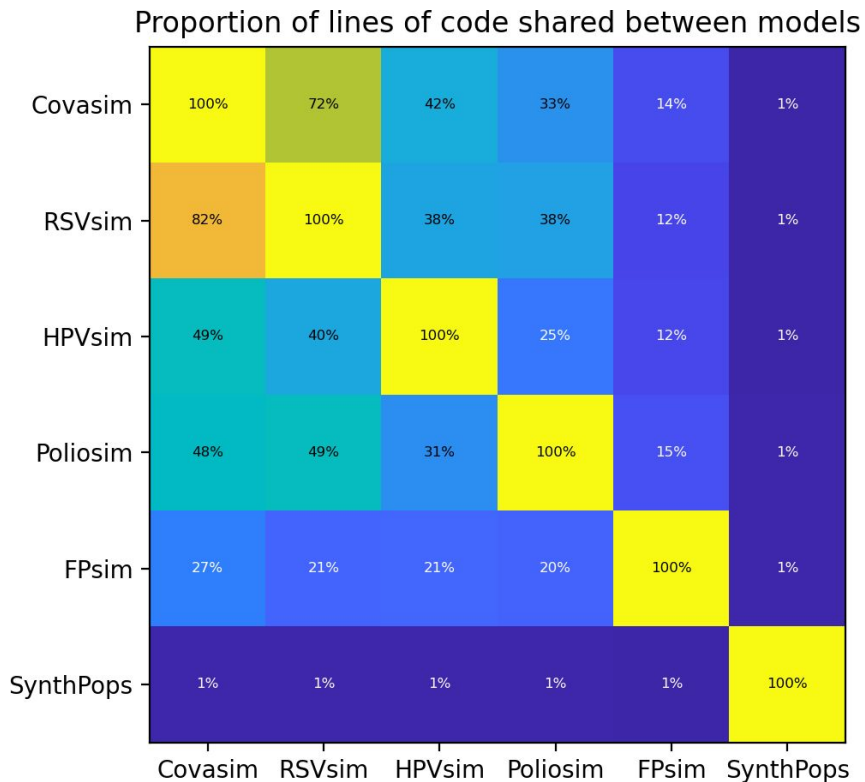
- **Agent-based model:** Simulates the state of individual people (agents) over time
- **Fundamental calculation:** the probability that a given agent, on a given time step, will change from one state to another, e.g.:
 - susceptible -> infected
 - undiagnosed -> diagnosed
 - ill -> dead
- Transmission across **networks**



From Covasim to *sim

- Mature (released and in use)
 - **Covasim** (COVID)
 - **HPVsim** (human papillomavirus)
 - **FPsim** (family planning)
 - **SynthPops** (population modeling)
- Alpha (undergoing testing)
 - **Poliosim** (poliomyelitis)
 - **STIsim**★ (all STIs + HIV)
- Pre-alpha (undergoing development)
 - **RSVsim**★ (respiratory syncytial virus)
 - **TBsim**★ (tuberculosis)
 - **Typhoidsim**★ (typhoid)

★ = *Implemented with the Starsim framework*



Design and implementation

Starsim architecture

Minimalist

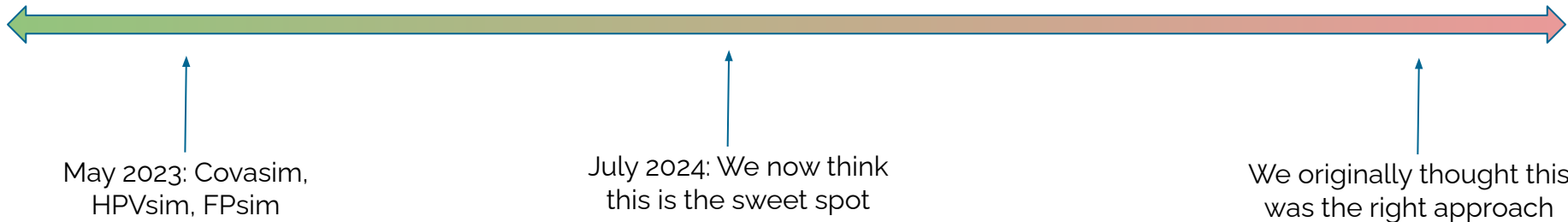
A loose set of design principles and norms that underlie a suite of dynamic epi models

Intermediate

An interlocking set of modules that provide core functions to different epi models

Maximalist

A single unified codebase for dynamic modeling that covers many different aspects of health



Starsim modular structure

Sim

People

States [alive, age, vaccinated, HPV infected, etc.]

Networks

Sexual

Risk-stratified

Preference

Age-mixed

Environmental

Respiratory

Household

Community

School

Workplace

Injecting

...

...

Custom

Demographics

Pregnancy

Death

Migration

Diseases (& health states)

HPV

TB

Diabetes

States

Results

HIV

Nutrition

Syphilis

Typhoid

+ Custom

Connectors

Disease interactions

Network interactions

Interventions

Vaccines

Diagnostics

Treatment

Analyzers

Calibration

Age binning

Trans. trees

Results

Plotting

Export

Vision for Starsim

Starsim: IDM's next-generation agent-based model framework

To durably promote quantitative decision making...

We seek to build inclusive, collaborative, and open modeling communities...

Via tools that encourage innovation, scientific curiosity, and rapid exploration.

We take a **user-centric approach** to software design:

1. All aspects should be simple, intuitive, and accessible → **easier** for ourselves and others
2. Software should get out of the way of science, with simple solutions preferred → facilitates **creativity and boldness** in which scientific/policy questions to tackle
3. Embrace diverse users, from novice to elite → we **cater to all levels** of technical expertise
4. Users trust us (rigor) and we trust them (power) → we **don't need to control** every use case

Starsim's design philosophy

"Common tasks should be simple"

e.g.

1. Defining parameters
2. Running a simulation
3. Plotting results

"...while uncommon tasks can't always be simple, but still should be possible"

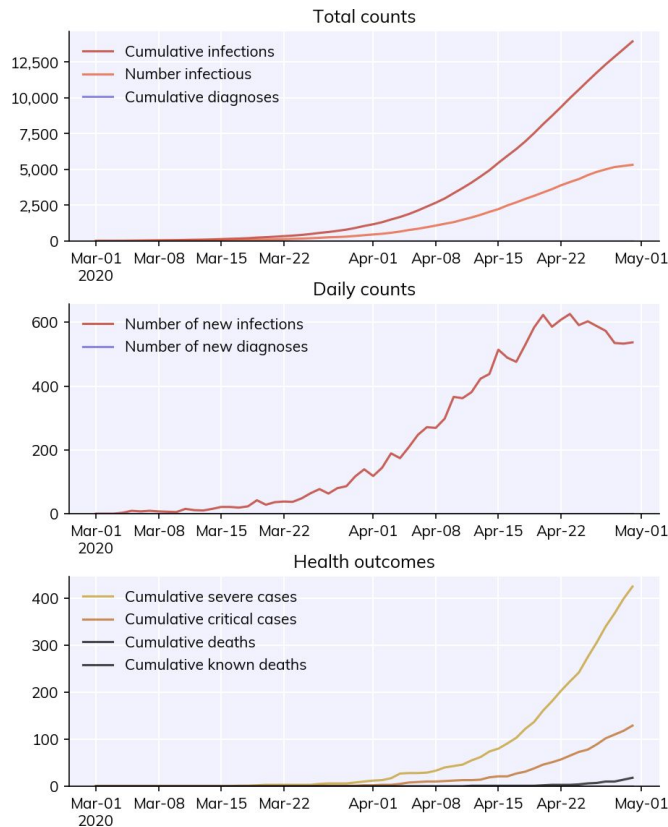
e.g.

1. Using a nonstandard distribution of viral load
2. Writing a custom goodness-of-fit function
3. Defining a new population structure

Code:

```
import covasim as cv
sim = cv.Sim()
sim.run()
sim.plot()
```

Output:



* One-line version:
`import covasim as cv; cv.Sim().run().plot()`

How **not** to write simulations in Python (or R)

- Slow implementation as "loop-de-loop":
 - On each timestep ($n \approx 700$):
 - For each person ($n \approx 30,000$):
 - For each state ($n \approx 30$):
 - » Update! (1 billion times)
- Conceptually simple, fine in C++, but very inefficient in Python/R

```
if self.alive: # Do not move through step if not alive

    self.age_person() # Age person in units of the timestep
    self.check_mortality() # Decide if person dies
    if not self.alive:
        return self.step_results

if self.sex == 0 and self.age < self.pars['age_limit_fecundity']:

    if self.pregnant:
        self.check_delivery() # Deliver with birth outcomes
        self.update_pregnancy() # Advance gestation in timestep
        if not self.alive:
            return self.step_results

    if not self.pregnant:
        self.check_sexually_active()
        if self.pars['method_age'] <= self.age < self.pars['age_lim_fecund']:
            self.update_contraception(t, y)
        self.check_lam()
        if self.sexually_active:
            self.check_conception() # Decide if conceives
        if self.postpartum:
            self.update_postpartum() # Updates postpartum counter

if self.lactating:
    self.update_breastfeeding()
```

A better way to write simulations in Python

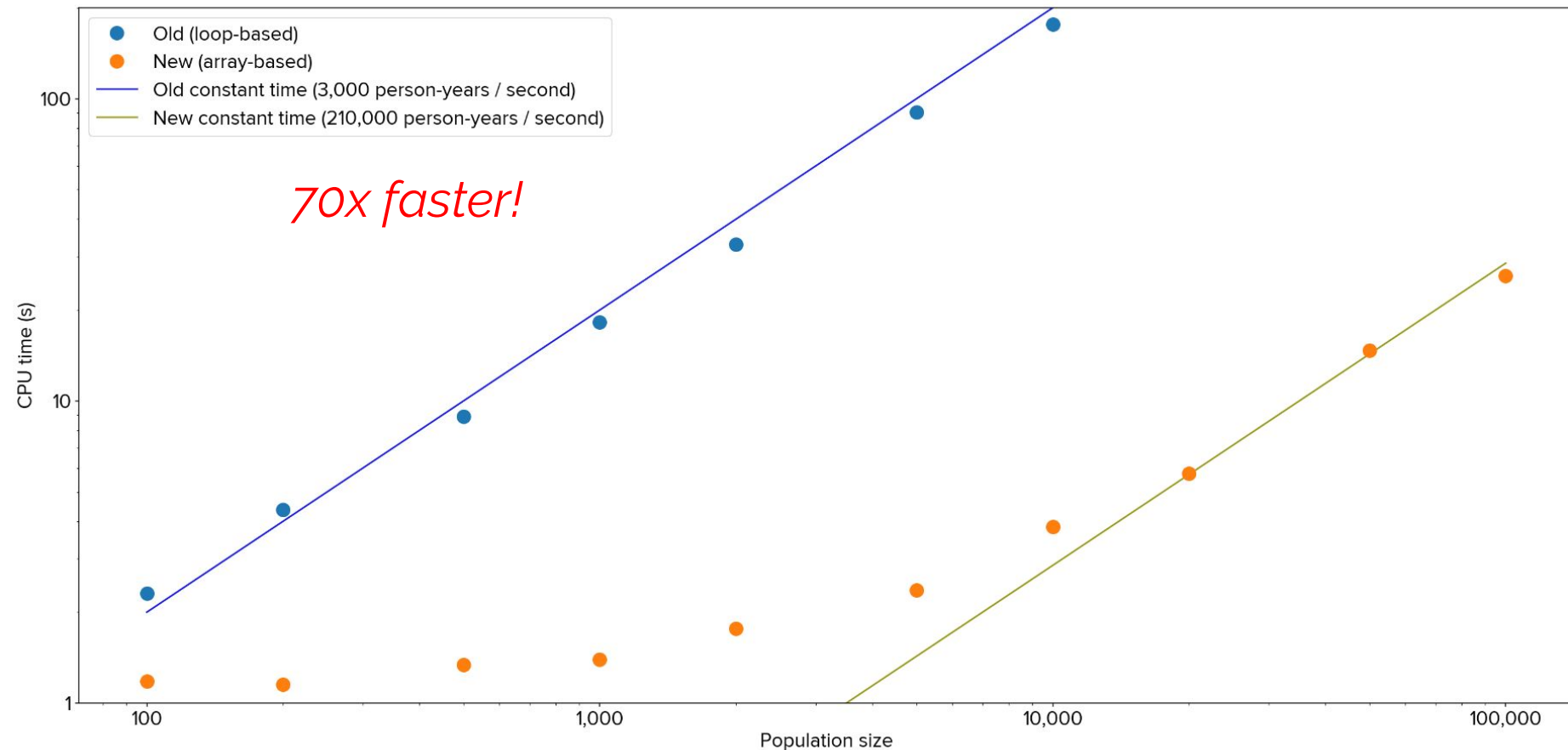
- Fast implementation via "arrayification":
 - On each timestep ($n \approx 700$):
 - For each state ($n \approx 30$):
 - Update everyone! (20,000 times)
- For loops become vector math, if statements become set operations
- Up to **~100x faster**: as fast as C++ (plus shorter code!)

```
alive = self.alive.uids # "uids" is "universal identifiers"
self.age_person(alive) # Age person in units of the timestep
self.check_mortality(alive) # Decide if person dies

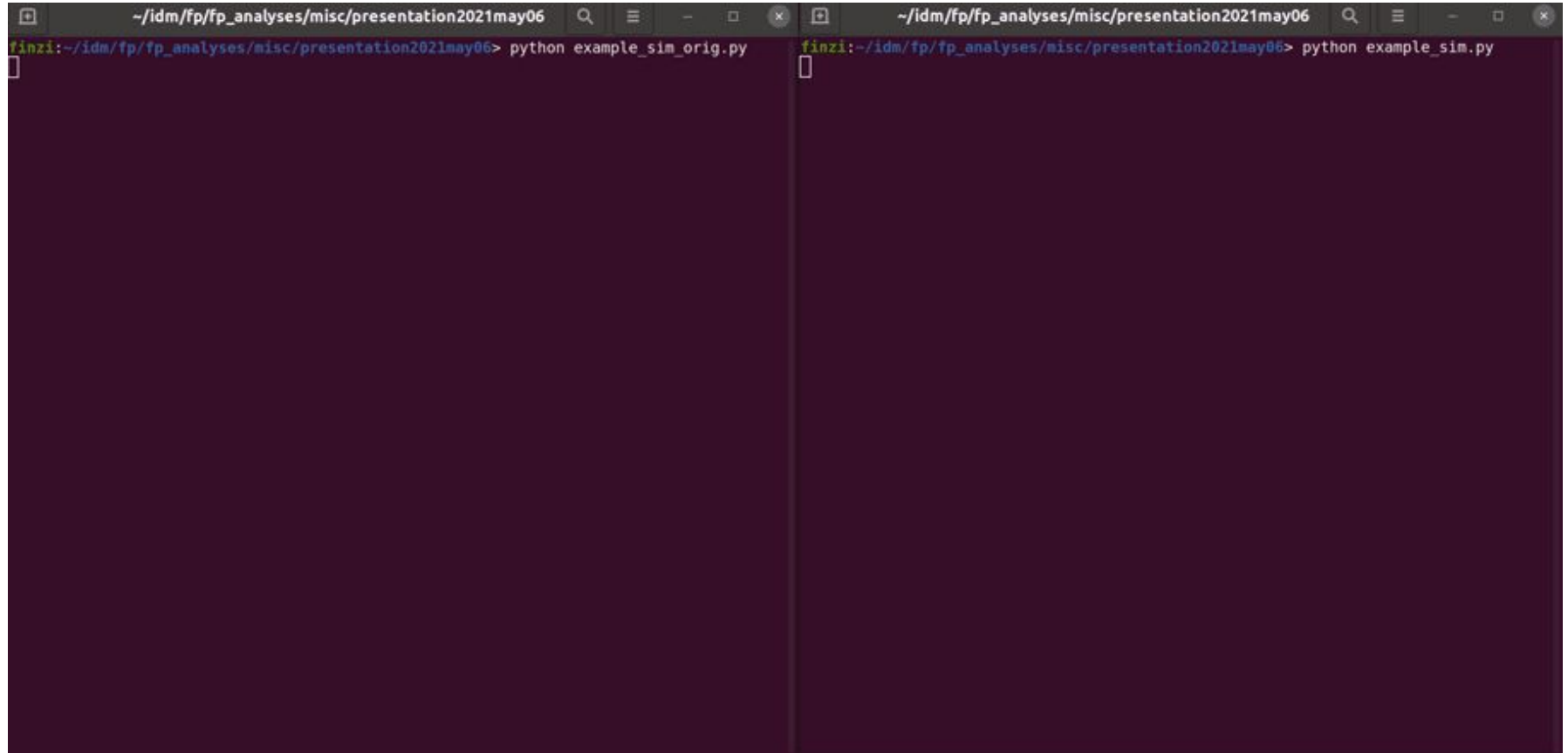
within_age = self.age < self.pars.age_lim_fecund # Within age limit
fecund = self.alive * (~self.sex) * within_age.uids # sex==0 is female
preg = self.pregnant.uids
nonpreg = fecund - preg # Find the set difference
lact = self.lactating.uids

# Update everything
self.check_delivery(preg) # Deliver with birth outcomes
self.update_pregnancy(preg) # Advance gestation in timestep
self.check_sexually_active(nonpreg)
self.update_contraception(nonpreg)
self.check_lam(nonpreg)
self.update_postpartum(nonpreg) # Updates postpartum counter
self.update_breastfeeding(lact)
self.check_conception(nonpreg) # Decide if conceives
```

How much difference does it make?



How much difference does it make?



The image shows two side-by-side terminal windows. Both windows have a title bar with the path `~/idm/fp/fp_analyses/misc/presentation2021may06`. The left window shows the command `python example_sim_orig.py` being executed. The right window shows the command `python example_sim.py` being executed. Both windows are currently empty, suggesting the scripts have not yet produced any output or the output is not visible in the current view.

```
~/idm/fp/fp_analyses/misc/presentation2021may06  
finzi:~/idm/fp/fp_analyses/misc/presentation2021may06> python example_sim_orig.py  
[ ]
```

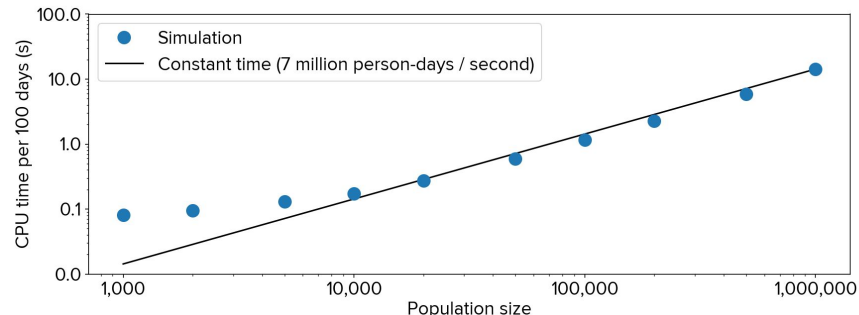
```
~/idm/fp/fp_analyses/misc/presentation2021may06  
finzi:~/idm/fp/fp_analyses/misc/presentation2021may06> python example_sim.py  
[ ]
```

Overall performance

- Implementation based on arrays (not objects) and Numba
- Pure Python, yet runs as fast as pure C/C++ models
- Feasible to run multiple realistic scenarios on standard laptop in minutes

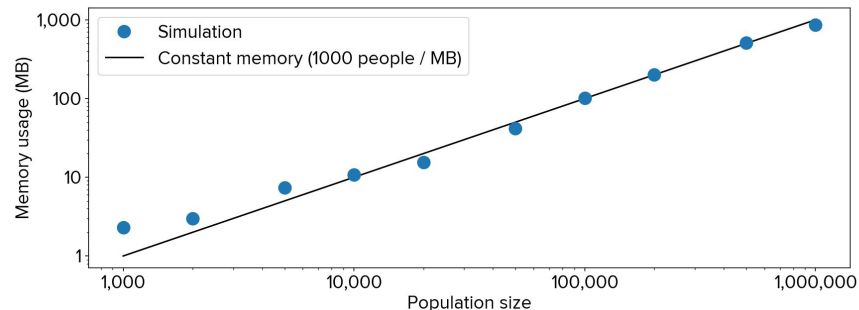
People (object-based)

Person A			Person B			Person C	
uid	23928		uid	41135		uid	76851
age	55.1		age	13.5		age	83.2
dead	0		dead	0		dead	1
susceptible	1		susceptible	0		susceptible	0
infected	0	...	infected	1	...	infected	1
diagnosed	0		diagnosed	0		diagnosed	1
...			
date_infected	NaN		date_infected	44		date_infected	46
date_diagnosed	NaN		date_diagnosed	NaN		date_diagnosed	53



People (array-based)

	Person A	...	Person B	...	Person C
uid (int)	23928	...	41135	...	76851
age (float)	55.1	...	13.5	...	83.2
dead (bool)	0	...	0	...	1
susceptible (bool)	1	...	0	...	0
infected (bool)	0	...	1	...	1
diagnosed (bool)	0	...	0	...	1
...
date_infected (float)	NaN	...	44	...	46
date_diagnosed (float)	NaN	...	NaN	...	53

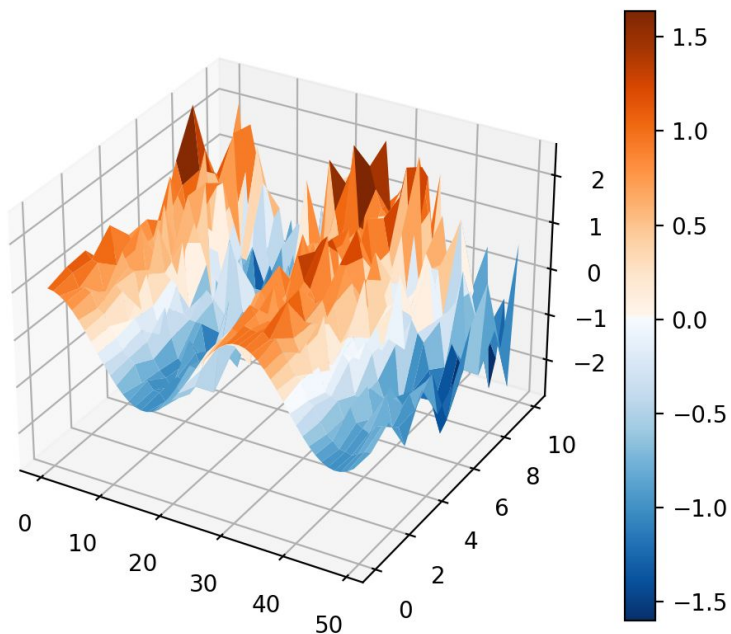


Powered by Sciris

Starsim uses [Sciris](#) to simplify common tasks



To make this:



Instead of this:

```
1 # Define random wave generator
2 import numpy as np
3
4 def randwave(std, xmin=0, xmax=10, npts=50):
5     np.random.seed(int(100*std)) # Ensure differences between runs
6     a = np.cos(np.linspace(xmin, xmax, npts))
7     b = np.random.randn(npts)
8     return a + b*std
9
10 # Other imports
11 - import time
12 - import multiprocessing as mp
13 - import pickle
14 - import gzip
15 - import matplotlib.pyplot as plt
16 - from mpl_toolkits.mplot3d import Axes3D # Unused but must be imported
17
18 # Start timing
19 - start = time.time()
20
21 # Calculate output in parallel
22 - multipool = mp.Pool(processes=mp.cpu_count())
23 - waves = multipool.map(randwave, np.linspace(0, 1, 11))
24 - multipool.close()
25 - multipool.join()
26
27 # Save to files
28 - filenames = []
29 - for i,wave in enumerate(waves):
30     - filename = f'wave{i}.obj'
31     - with gzip.GzipFile(filename, 'wb') as fileobj:
32         - fileobj.write(pickle.dumps(wave))
33     - filenames.append(filename)
34
35 # Create dict from files
36 - data_dict = {}
37 - for fname in filenames:
38     - with gzip.GzipFile(fname) as fileobj:
39         - filestring = fileobj.read()
40     - data_dict[fname] = pickle.loads(filestring)
41
42 # Create 3D plot
43 - data = np.array([data_dict[fname] for fname in filenames])
44 - fig = plt.figure()
45 - ax = plt.axes(projection='3d')
46 - ny,nx = np.array(data).shape
47 - x = np.arange(nx)
48 - y = np.arange(ny)
49 - X, Y = np.meshgrid(x, y)
50 - surf = ax.plot_surface(X, Y, data, cmap='coolwarm')
51 - fig.colorbar(surf)
52
53 # Print elapsed time
54 - elapsed = time.time() - start
55 - print(f'Elapsed time: {elapsed:0.1f} s')
```

Do this:

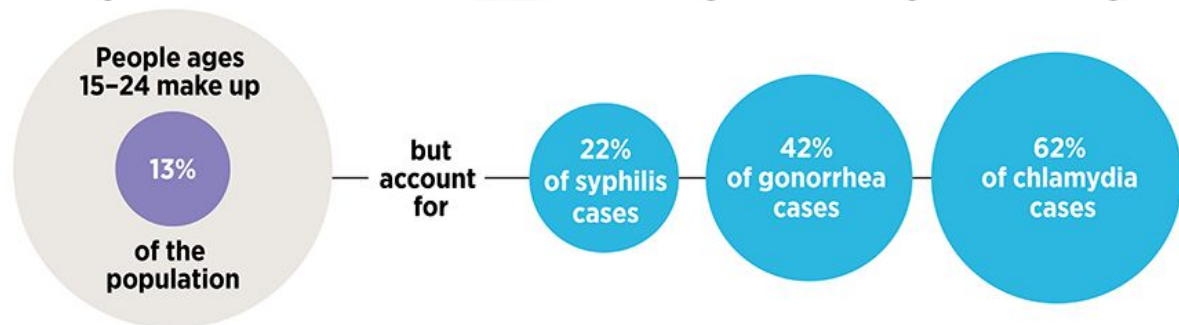
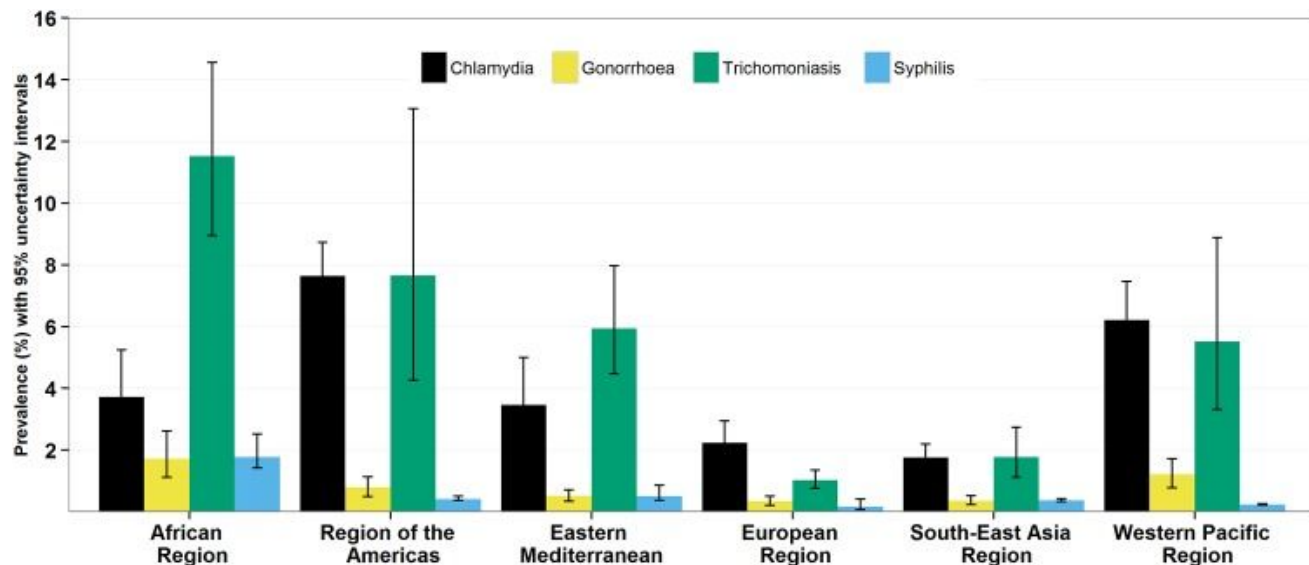
```
1 # Define random wave generator
2 import numpy as np
3
4 def randwave(std, xmin=0, xmax=10, npts=50):
5     np.random.seed(int(100*std)) # Ensure differences between runs
6     a = np.cos(np.linspace(xmin, xmax, npts))
7     b = np.random.randn(npts)
8     return a + b*std
9
10 # Other imports
11 + import sciris as sc
12
13 # Start timing
14 + T = sc.timer()
15
16 # Calculate output in parallel
17 + waves = sc.parallelize(randwave, np.linspace(0, 1, 11))
18
19 # Save to files
20 + filenames = [sc.save(f'wave{i}.obj', wave) for i,wave in enumerate(waves)]
21
22 # Create dict from files
23 + data = sc.odict({fname:sc.load(fname) for fname in filenames})
24
25 # Create 3D plot
26 + sc.surf3d(data[:,], cmap='orangeblue')
27
28 # Print elapsed time
29 + T.toc()
```

Application: Modeling co-circulating STIs

Global burden of STIs

- >1 million new STIs every day
- >1 million pregnant women with syphilis in 2016; ~350,000 adverse birth outcomes
- Burden varies by country/age/sex

Global STI prevalence from [Neman et al \(2012\)](#)

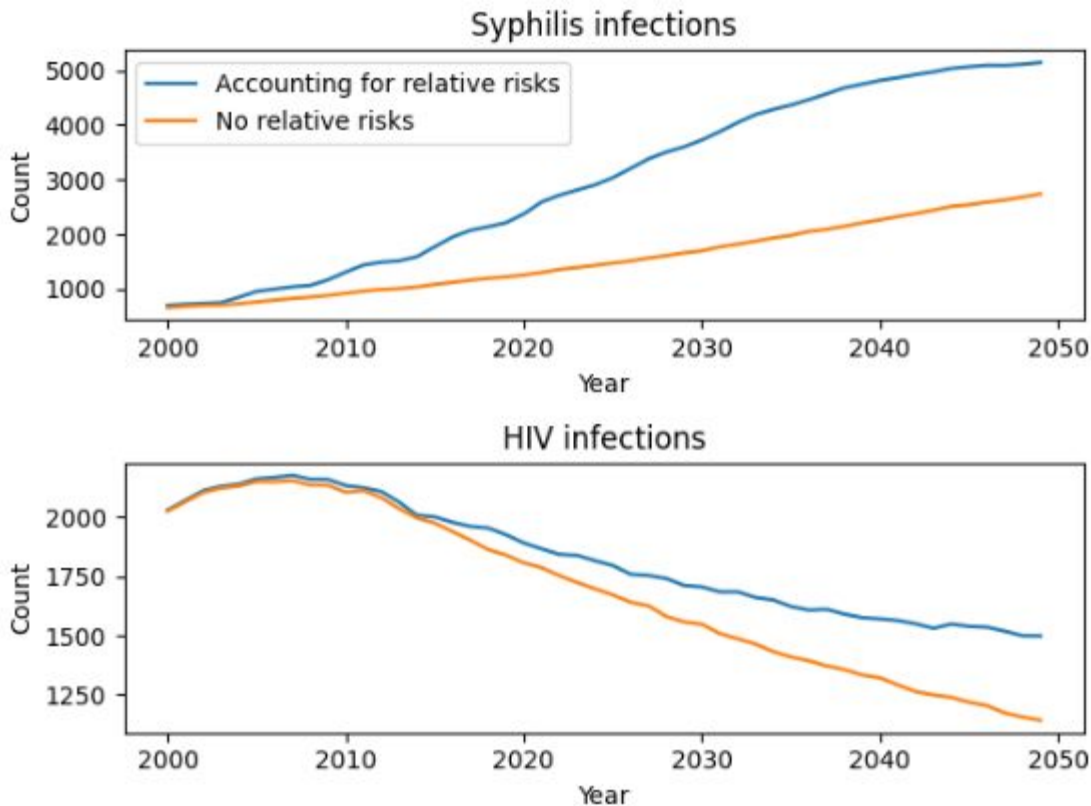


Example policy questions

- How many DALYs would be averted by improved syphilis diagnostics?
 - Better syphilis control → better HIV control: combined DALY impact
- What is the **optimal combination** of **STI diagnostics** and treatments to offer via antenatal clinics to minimize negative **birth outcomes**?
 - Need to consider **cross-cutting interventions**
 - Must capture **co-transmission**
- Which populations are contributing most to **onward transmission** of STIs, and should be **prioritized** for testing and treatment?
 - Need to capture **transmission pathways** and **network effects**

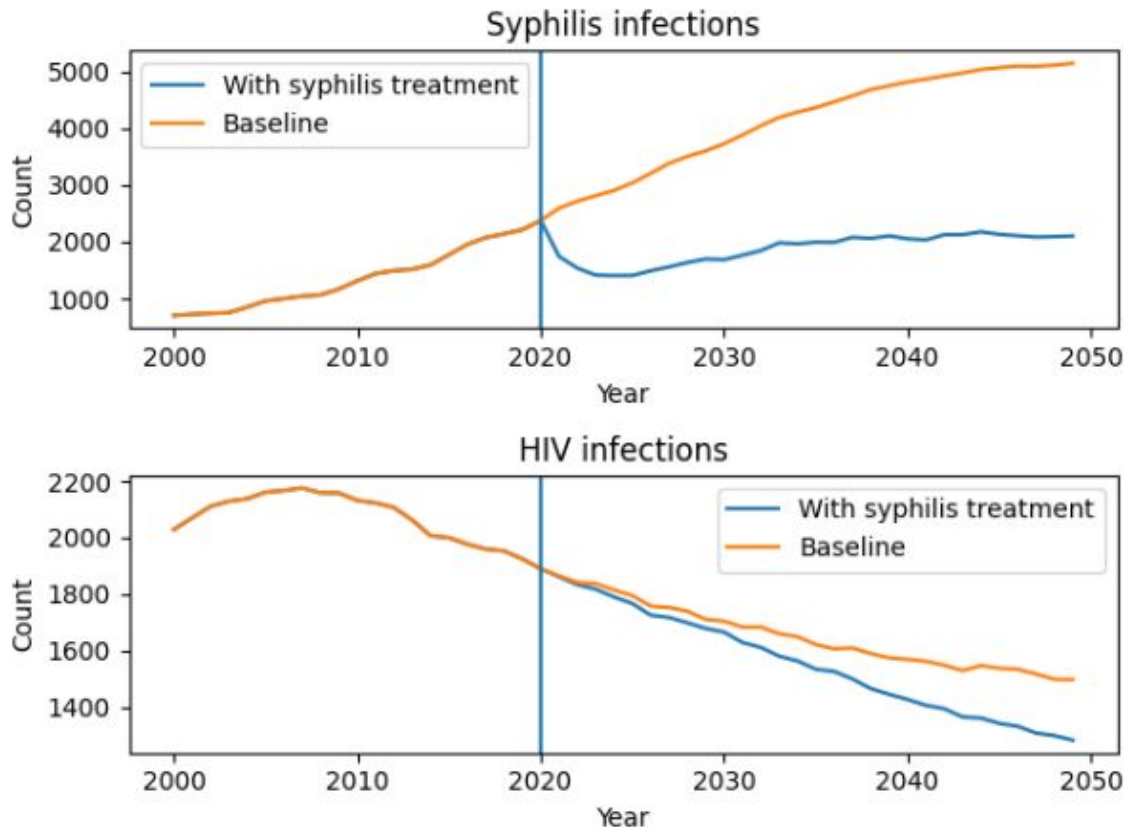
Modeling HIV and syphilis

- Syphilis infection increases the risk of HIV acquisition (~2.7x rel risk)
- Diagnosis of early syphilis carries a high risk of consequent HIV
- Accounting for these effects, we see increased transmission



Modeling syphilis treatment and HIV

- Syphilis treatment can moderate HIV transmission
- Modeling the secondary effects of treatment for one disease on burden of related diseases captures the true impact



Summary

Limitations and discussion

- Have not spoken about calibration or validation, critical components of turning a modeling framework into a model
 - Starsim has built-in calibration methods, but your mileage may vary
- Starsim is still very much in development, please give us feedback and we welcome any and all collaboration!
- Underlying philosophy drove practical user-centric design decisions
- Next phases: Applications, ongoing refinement
- Open question: Should we think of Starsim as a general-purpose agent-based model?

Starsim access

- **Code:**

- Starsim: <https://starsim.org/>
- Covasim: <http://covasim.org>
- FPsim: <http://fpsim.org>
- HPVsim: <http://hpvsim.org>

- **Documentation/tutorials:**

- Starsim: <http://docs.starsim.org>
- Covasim: <http://docs.covasim.org>
- FPsim: <http://docs.fpsim.org>
- HPVsim: <http://docs.hpvsim.org>

- **Methods papers:**

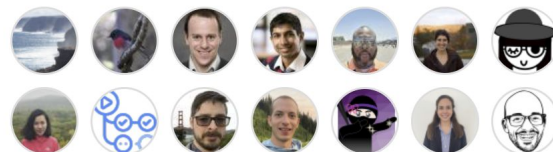
- Covasim: Kerr et al 2021, [PLoS CB](#)
- FPsim: O'Brien et al 2023, [npj Women's Health](#)
- HPVsim: Stuart et al 2024, [PLoS CB](#)

Acknowledgements

Starsim model development has been led by Cliff Kerr, Robyn Stuart, Romesh Abeysuriya, Paula Sanz-Leon, and Daniel Klein.

Numerous experts contributed to the development of individual disease models, including Michelle O'Brien, Evans Omondi, Sharon Achilles, Maike Scharp, Darcy Rao, Haina Shin, Celina Schocken, Peter Dull, John Schiller, Chris Karp, Holger Kanzler, Michael Marks, Mark Jit, Marc Brisson, Jasmina Panovska-Griffiths, and many others.

Contributors 19



Thank you!