Covasim: an agent-based model of COVID-19 dynamics and interventions

Cliff C. Kerr[1*], Robyn M. Stuart[2,3†], Dina Mistry[1†], Romesh G. Abeysuriya[3], Katherine Rosenfeld[1], Gregory R. Hart[1], Rafael C. Núñez[1], Jamie A. Cohen[1], Prashanth Selvaraj[1], Brittany Hagedorn[1], Lauren George[1], Michał Jastrzębski[4], Amanda Izzo[1], Greer Fowler[1], Anna Palmer[3], Dominic Delport[3], Nick Scott[3], Sherrie Kelly[3], Caroline S. Bennette[1], Bradley Wagner[1], Stewart Chang[1], Assaf P. Oron[1], Edward Wenger[1], Jasmina Panovska-Griffiths[5,6] , Michael Famulare[1], and Daniel J. Klein[1]

[1] *Institute for Disease Modeling, Global Health Division, Bill & Melinda Gates Foundation, Seattle, WA, USA*
[2] *Department of Mathematical Sciences, University of Copenhagen, Copenhagen, Denmark*
[3] *Burnet Institute, Melbourne, VIC, Australia*
[4] *GitHub, Inc., San Francisco, CA, USA*
[5] *Department of Applied Health Research, University College London, London, UK*
[6] *Wolfson Centre for Mathematical Biology and The Queen's College, Oxford University, Oxford, UK*

** Corresponding author. E-mail: ckerr@idmod.org*
† Contributed equally

Abstract

The COVID-19 pandemic has created an urgent need for models that can project epidemic trends, explore intervention scenarios, and estimate resource needs. Here we describe the methodology of Covasim (COVID-19 Agent-based Simulator), an open-source model developed to help address these questions. Covasim includes country-specific demographic information on age structure and population size; realistic transmission networks in different social layers, including households, schools, workplaces, long-term care facilities, and communities; age-specific disease outcomes; and intrahost viral dynamics, including viral-load-based transmissibility. Covasim also supports an extensive set of interventions, including non-pharmaceutical interventions, such as physical distancing and protective equipment; pharmaceutical interventions, including vaccination; and testing interventions, such as symptomatic and asymptomatic testing, isolation, contact tracing, and quarantine. These interventions can incorporate the effects of delays, loss-to-follow-up, micro-targeting, and other factors. Implemented in pure Python, Covasim has been designed with equal emphasis on performance, ease of use, and flexibility: realistic and highly customized scenarios can be run on a standard laptop in under a minute. In collaboration with local health agencies and policymakers, Covasim has already been applied to examine epidemic dynamics and inform policy decisions in more than a dozen countries in Africa, Asia-Pacific, Europe, and North America.

1 Introduction

More than a year after COVID-19 was first identified, governments continue to be faced with an urgent need to understand the rapidly evolving pandemic landscape and translate it into policy. Since the onset of the pandemic, mathematical modeling has been at the heart of informing this decision-making. Numerous statistical models and data visualization tools have been developed over the last year in an attempt to meet this demand, with varying purposes, structures, and levels of detail and complexity; for example, despite their limitations (1), data dashboards have proven crucial for understanding the current state of the epidemic on both global and local scales (2,3). However, more detailed models are needed to evaluate scenarios based on complex intervention strategies. These strategies are important to evaluate in order to understand the epidemiological impact of reopening schools, businesses, and society.

Models for examining COVID-19 transmission and control measures can be broadly divided into two main types: compartmental models and agent-based models (also called individual-based or microsimulation models), with the former generally being simpler and faster, while the latter are generally more complex, detailed, and computationally expensive. Numerous compartmental models have been developed or repurposed for COVID-19: Walker et al. (4) used an age-structured stochastic "susceptible, exposed, infectious, recovered" (SEIR) model to determine the global impact of COVID-19 and the effects of various social distancing interventions; Read et al. (5) developed an SEIR model to estimate the basic reproduction number in Wuhan; Keeling et al. (6) used one to look at the efficacy of contact tracing as a containment measure; and Dehning et al. (7) used an SIR model to quantify the impact of intervention measures in Germany. In models such as those by Giordano et al. (8) and Zhao and Chen (9), compartments are further divided to provide more nuance in simulating progression through different disease states, and have been deployed to study the effects of various population-wide interventions such as social distancing and testing on COVID-19 transmission.

For microsimulation models, several agent-based influenza pandemic models have been repurposed to simulate the spread of COVID-19 transmission and the impact of social distancing measures in the United Kingdom (10), Australia (11), Singapore (12), and the United States (13). Additionally, new agent-based models have been developed to evaluate the impact of social distancing and contact tracing (14–18) and superspreading (19). Features of these models include accounting for the number of household and non-household contacts (13,15,16); the age and clustering of contacts within households (13,14,16); and the microstructure in schools and workplace settings informed by census and time-use data (14). Branching process models have also been used to investigate the impact of non-pharmaceutical intervention strategies (20,21) and the proportion of unobserved infections (22).

In developing Covasim, our aim was to produce a tool that would be capable of informing real-world policy decisions for a variety of national and subnational settings. We wanted to capture the benefits of agent-based modeling (in particular, the ability of such models to simulate the kinds of microscale policies being used to respond to the COVID-19 pandemic), whilst making use of recent advances in software tools and computational methods to minimize the complexity and computation time typically associated with such models. In this regard, Covasim is most similar to the OpenABM-Covid19 model (23,24), which has also been developed as a high-performance, user-friendly, general-purpose COVID model.

Overall, the design principle we followed with Covasim was to make common usage patterns as simple as possible, while still giving the user the ability to customize virtually all aspects of the simulation. For example, Covasim comes pre-loaded with demographic data for each country (Section 2.4), but users can also define custom populations and contact networks down to the level of a single city (25) or even university (26). In addition, Covasim includes six built-in interventions (Section 2.5), but custom interventions of arbitrary complexity can also be defined (Fig. 9). In addition, Covasim's high performance for an agent-based model, achieved via dynamic rescaling (Section 2.6.2) and array-based computations (Section 2.7.1), means that most analyses can be run on a standard laptop, removing the need to use a high-performance computing cluster except for large parameter sweeps or model calibrations (Section 2.6.8). These design choices are intended to allow users to start running simple Covasim analyses quickly, while providing flexibility later if more detailed data become available or if the modeling questions become more nuanced.

To date, Covasim has been used by researchers and public health officials in over a dozen countries. Covasim has been used to inform policy decisions in the United States (25,27), Vietnam (28), the United Kingdom (29), and Australia (30). It has also been used for research studies in these locations (31–34), and well as other countries including India, Russia, Kenya, and South Africa. This paper describes the methodology underlying Covasim, and provides several examples illustrating its use, including an application to Seattle where Covasim scenarios were used to inform a rapid policy decision, with subsequent validation of these findings by real-world data.

2 Methods

2.1 Overview

Covasim simulates the state of individual people, known as agents, over a number of discrete time steps. Conceptually, the model is largely focused on a single type of calculation: the probability that a given agent on a given time step will change from one state to another, such as from susceptible to infected, or from critically ill to dead. Once these probabilities have been calculated, a pseudorandom number generator with a user-specified seed is used to determine whether the transition actually takes place for a given model run.

The logical flow of a single Covasim run is as follows. First, the simulation object is created, then the parameters are loaded and validated for internal consistency, and any specified data files are loaded (described in Section 2.6.1). Second, a population of agents is created, including age, sex, and comorbidities for each agent, drawing from location-specific data distributions where available; the, agents are then connected into contact networks based on their age and other statistical properties (Section 2.4). Next, the integration loop begins. On each timestep (which corresponds to a single day by default), the order of operations is: dynamic rescaling (Section 2.6.2); applying health system constraints (Section 2.6.3); updating the state of each agent, including disease progression (Section 2.2); importation events (Section 2.6.4); applying interventions (Section 2.5); calculating disease transmission events across each infectious agent's contact network (Section 2.3); collating outputs into results arrays (Section 2.6.5); and applying analyzers (Section 2.6.7). The following sections describe each step in more detail.

2.2 Disease progression

In Covasim, each individual is characterized as either susceptible, exposed (i.e., infected but not yet infectious), infectious, recovered, or dead, with infectious individuals additionally categorized according to their symptoms: asymptomatic, presymptomatic, mild, severe, or critical. A schematic diagram of the model structure is shown in Fig. 1.

The length of time after exposure before an individual becomes infectious is set by default to be a log-normal distribution with a mean of 4.6 days, which is within the range of values reported across the literature (Table 1). The length of time between the start of viral shedding and symptom onset is assumed to follow a log-normal distribution with a mean of 1 day (Table 1). Exposed individuals may develop symptoms or may remain asymptomatic. Individuals with symptoms are disaggregated into either mild, severe, or critical cases, with the probability of developing a more acute case increasing with age (Table 2). Covasim can also model the effect of comorbidities, which act by modifying an individual's probability of developing severe symptoms (and hence critical symptoms and death). By default, comorbidity multipliers are set to 1 since they are already factored into the marginal age-dependent disease progression rates.

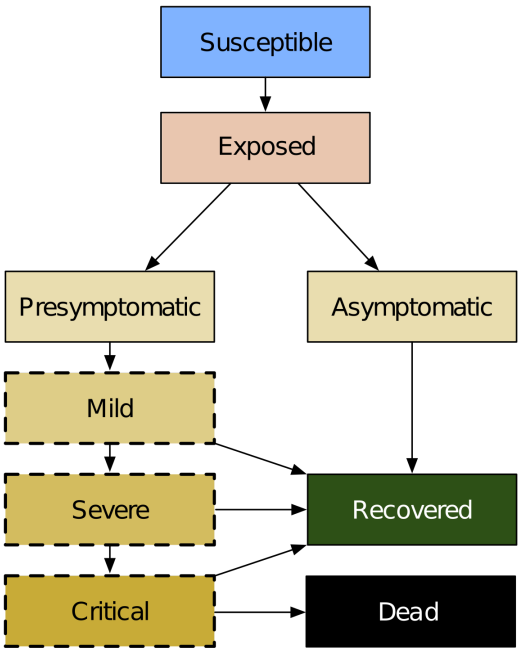


Fig. 1: Covasim model structure, including infection (exposure), disease progression, and final outcomes. Yellow shading indicates that an individual is infectious and can transmit the disease to other susceptible agents. States with a dashed border are considered symptomatic with respect to symptomatic versus asymptomatic testing.

Estimates of the duration of COVID-19 symptoms and the length of time that viral shedding occurs are highly variable, but durations are generally reported to increase according to acuity (35,36). We reflect this in our model with different recovery times for asymptomatic individuals, those with mild symptoms, and those with severe symptoms, as summarized in Table 1. All non-critical cases are assumed to recover, while critical cases either recover or die, with the probability of death increasing with age (Table 2).

Table 1. Default duration parameters, in days, used in the Covasim model.

Parameter	**Description**	**Distribution** (mean, std)	**Source**
τ_{inf}	Exposed-to-infectious: Length of time after exposure before an individual is infectious (i.e., has begun viral shedding)	τ_{inf} ~ *lognormal(4.5, 1.5)*	From Lauer et al. (37); additional sources Du et al., Nishiura et al., and Pung et al. (38–40).
τ_{sym}	Infectious-to-symptomatic: Length of time after viral shedding has begun before an individual has symptoms	τ_{sym} ~ lognormal(1.1, 0.9)	Linton et al. (41) report the incubation period as 5.6 days (95% CI: 5.0–6.3 days). Using the period of exposure before becoming infectious, we infer the period of viral shedding before symptomatic. However, other studies have estimated longer periods, e.g. (42).
τ_{sev}	Symptomatic-to-severe: Length of time after symptoms have started that symptoms have become severe (i.e., requiring hospitalization)	τ_{sev} ~ lognormal(6.6, 4.9)	Linton et al. (41), Wang et al. (43)
τ_{cri}	Severe-to-critical: Length of time after severe symptoms have started that symptoms have become critical (i.e., requiring ICU)	τ_{cri} ~ lognormal(1.5, 2.0)	Chen et al. (44), Wang et al. (43)
τ_{dea}	Critical-to-death: Time from onset of critical symptoms to death	τ_{dea} ~ lognormal(10.7, 4.8)	Verity et al. (45)
τ_{ra}	Time from infectiousness onset to recovery for asymptomatic cases	τ_{ra} ~ lognormal(8.0, 2.0)	Wölfel et al. (46)
τ_{rm}	Time from symptom onset to recovery for mild cases	τ_{rm} ~ lognormal(8.0, 2.0)	Wölfel et al. (46)
τ_{rs}	Time from onset of severe symptoms to recovery for severe cases	τ_{rs} ~ lognormal(18.1, 6.3)	Verity et al. (45)
τ_{rc}	Time from onset of critical symptoms to recovery for critical cases	τ_{rc} ~ lognormal(18.1, 6.3)	Verity et al. (45)

Table 2: Age-linked disease susceptibility, progression, and mortality probabilities. Key: r_{sus}: relative susceptibility to infection; p_{sym}: probability of developing symptoms; p_{sev}: probability of developing severe symptoms (i.e., sufficient to justify hospitalization); p_{cri}: probability of developing into a critical case (i.e., sufficient to require ICU); p_{dea}: probability of death (i.e., infection fatality ratio). Relative susceptibility values are derived from odds ratios presented in Zhang et al. (47). Mortality rates are based on O'Driscoll et al. (48) for ages <90 and Brazeau et al. (49) for ages >90. All other values are derived from Verity et al. (45) and Ferguson et al. (50), which did not differentiate 80-89 and 90+. Values were validated from model fits to data on numbers of cases, numbers of people hospitalized and in intensive care, and numbers of deaths from Washington and Oregon states. Note that "overall" values depend on the age structure of the population being modeled. For a population like the USA or UK, the symptomatic proportion is roughly 70%, while for populations skewed towards younger ages, this proportion is lower. Similarly, overall mortality rates are estimated to vary from 0.2% in Kenya to 0.9% in the USA and 1.4% in Italy.

	0-9	**10-19**	**20-29**	**30-39**	**40-49**	**50-59**	**60-69**	**70-79**	**80-89**	**90+**	**Overall**
r_{sus}	0.34	0.67	1.00	1.00	1.00	1.00	1.00	1.24	1.47	1.47	1.00
p_{sym}	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.90	0.5–0.75

p_{sev}	0.00050	0.00165	0.00720	0.02080	0.03430	0.07650	0.13280	0.20655	0.24570	0.24570	0.1–0.2
p_{cri}	0.00003	0.00008	0.00036	0.00104	0.00216	0.00933	0.03639	0.08923	0.17420	0.17420	0.05–0.1
p_{dea}	0.00002	0.00002	0.00010	0.00032	0.00098	0.00265	0.00766	0.02439	0.08292	0.16190	0.002–0.015

2.3 Transmission and within-host viral dynamics

Whenever a susceptible individual comes into contact with an infectious individual on a given day, transmission of the virus occurs according to probability β. For a well-mixed population where each individual has an average of 20 contacts per day, a value of β = 0.016 corresponds to a doubling time of roughly 4–6 days and an R_0 of approximately 2.2–2.7, with the exact value depending on the population size, age structure, and other factors. The value of β = 0.016 that is currently used as the default in Covasim was based on calibrations to data from Washington and Oregon states. However, this default value is too low for high-transmission contexts such as New York City or Lombardy (51), and may be too high for low-transmission contexts such as India (52). Hence, this parameter must be calibrated by the user to match local epidemic data, as described in Section 2.6.8.

If realistic network structure (i.e., households, schools, workplaces, and community contacts) is included, the value of β depends on the contact type. Default transmission probabilities are roughly 0.050 per contact per day for households, 0.010 for workplaces and schools, and 0.005 for the community. These values correspond to relative weightings of 10:2:2:1, chosen (a) for consistency with both time-use surveys (53) and studies of infections with known contact types (47), and (b) to have a weighted mean close to the default β value of 0.016 for a well-mixed population (i.e., if different network layers are not used). When combined with the default number of contacts in each layer, age-based susceptibility, and other factors, for a typical (unmitigated) transmission scenario, the proportions of transmission events that occur in each contact layer in the absence of interventions are approximately 30% via households, 25% via workplaces, 15% via schools, and 30% via the community. The value of β can also be modified by interventions, such as physical distancing, as described below.

In addition to allowing individuals to vary across disease severity and time spent in each disease state, we allow individual infectiousness to vary between people and over time. We use individual viral load to model these differences in infectivity. Several groups have found that viral load is highest around or slightly before symptom onset, and then falls monotonically (54–58). As a simple approximation to this viral time course, we model two stages of viral load: an early high stage followed by a longer low stage. By default, we set the viral load of the high stage to be twice as high as the low stage and to last for either 30% of the infectious duration or 4 days, whichever is shorter. The default viral load for each agent is drawn from a negative binomial distribution with mean 1.0 and shape parameter 0.45, which was the value most consistent with both international estimates (59,60) and fits to data in Washington state and Oregon. The daily viral load is used to adjust the per-contact transmission probability (β) for an agent on a given day. Viral loads for a representative sample of individuals given default parameter values are shown in Fig. 2. The proportion of transmissions by asymptomatic, presymptomatic, and symptomatic individuals varies by context; estimated proportions for Seattle are shown in (25).

Evidence is mixed as to whether transmissibility is lower if the infectious individual does not have symptoms (55). We take a default assumption that it is not, but include a parameter that can be modified as needed depending on the modeling application or context, noting that some studies have used much lower rates of infectiousness for asymptomatic individuals (61).

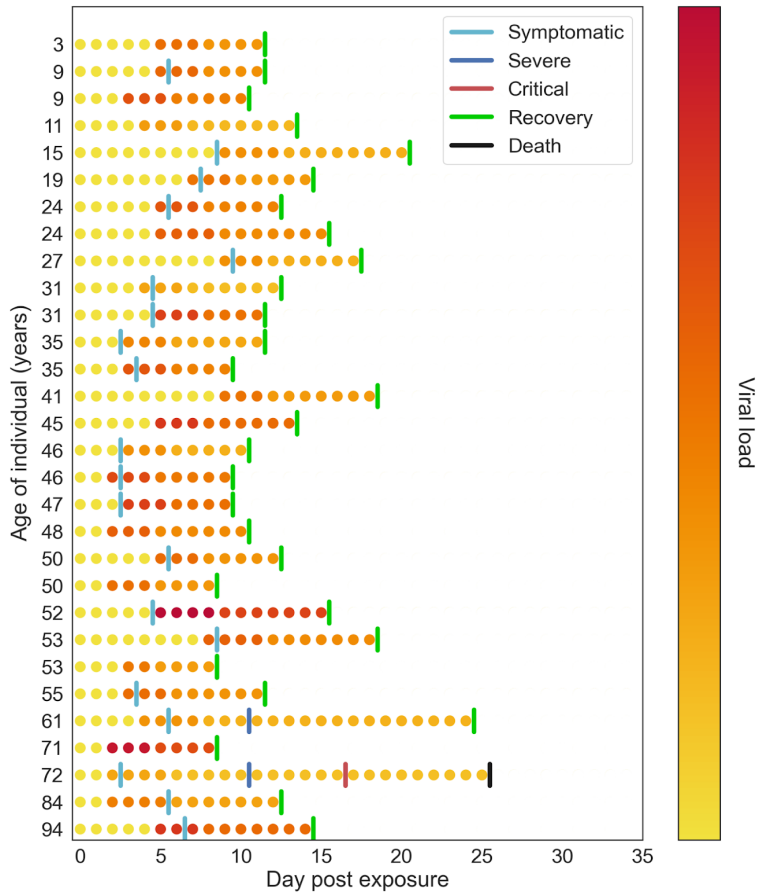


Fig. 2: Example of within-host viral load dynamics in Covasim. Each row shows a different agent in the model. Color indicates viral load, which typically peaks the day before or the day of symptom onset, before declining slowly.

2.4 Contact network models

Covasim is capable of generating and using three alternative types of contact networks: random networks, SynthPops networks, and hybrid networks. Each of these may be useful in different settings, and in addition users have the option of defining their own networks. Covasim's default contact networks are shown schematically in Fig. 3; different options for construction these networks are provided in the following sections. To facilitate easy adaptation to different contexts, Covasim comes pre-loaded with data on country age distributions and household sizes as reported by the UN Population Division 2019 (population.un.org).

2.4.1 Random networks

Covasim generates random networks by assuming that each person in the modeled population can come into contact with anyone else in the population. Each person is assigned a number of daily contacts, which is drawn from a Poisson distribution whose mean value can be specified by the user depending on the modeling context (with a default value of 20). The user can also decide whether these contacts should remain the same throughout the simulation, or whether they should be sampled randomly from the population each day.

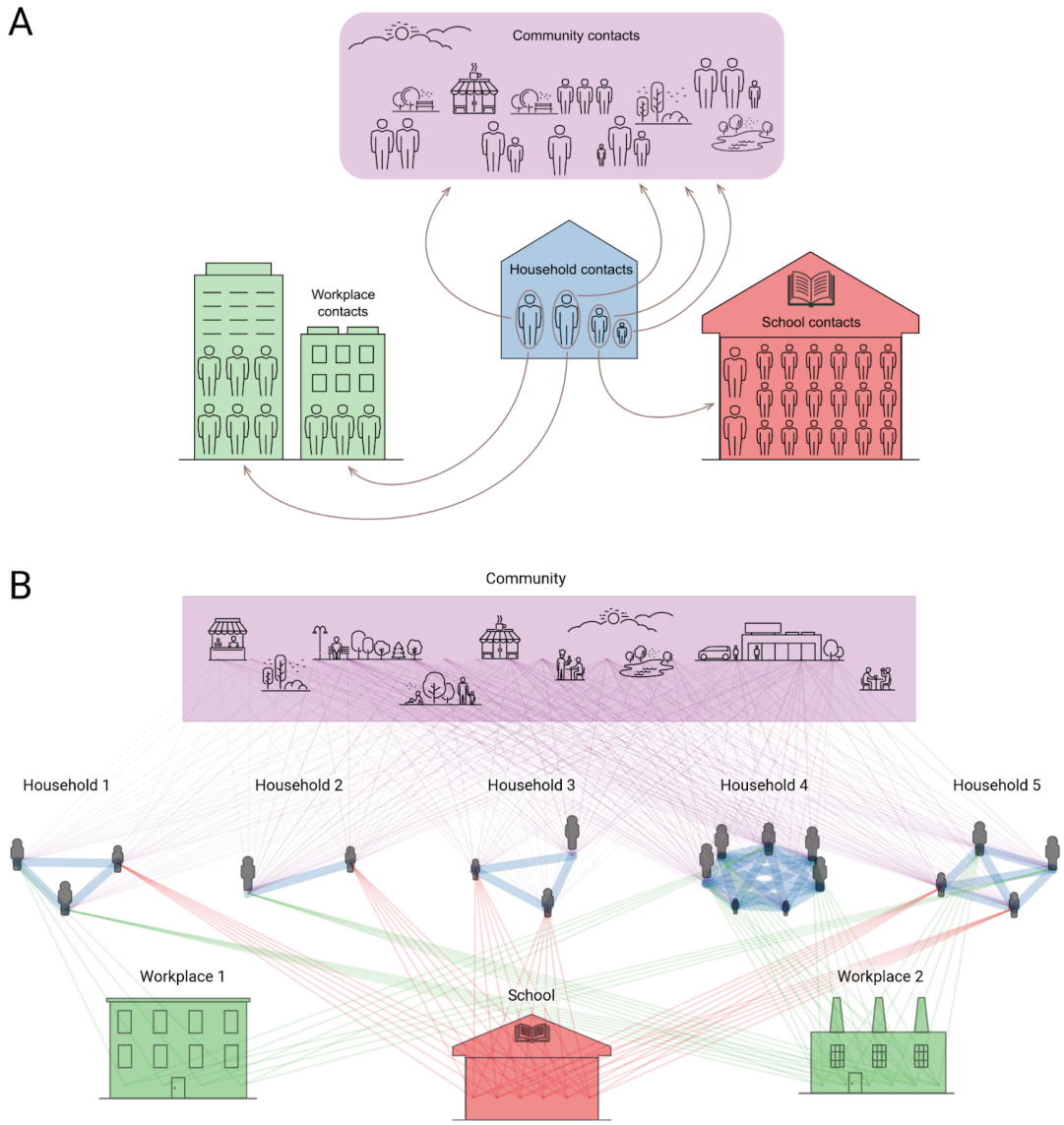


Fig. 3: Illustration of contact networks with multiple layers in Covasim. (A) In reality, individual people move between household, school, workplace, and community contact layers during the day. (B) In the model (shown here with a population of 20 people, with age structure and household sizes based on Malawi data), these dynamic contacts are approximated as static average daily contacts between layers. Individuals have different numbers of connections (lines) and connection weights (line widths; default relative weights shown) for each layer.

2.4.2 SynthPops networks

Covasim is integrated with SynthPops, an open-source data-driven model capable of generating realistic synthetic contact networks for populations; further information, including documentation and source code, is available from synthpops.org. Briefly, the method draws upon previously published models and empirical studies to infer high-resolution age-specific contact patterns in key settings (e.g., households, schools, workplaces, and the general community) relevant to the transmission of infectious diseases (62–64). Census or survey data such as those from Demographic and Health Surveys (65,66) are used by SynthPops to inform demographic characteristics (e.g., age, household size, school enrollment, and employment rates). Age-specific contact matrices, such as those in (62,67–69), are then used to generate individuals and their expected contacts in a multilayer network framework. By default, SynthPops generates household, school, and work contact networks; community connections are

generated using the random approach described above, and long-term care facilities can be included if data are available. An example synthetic network as generated by SynthPops is shown in Fig. 4.

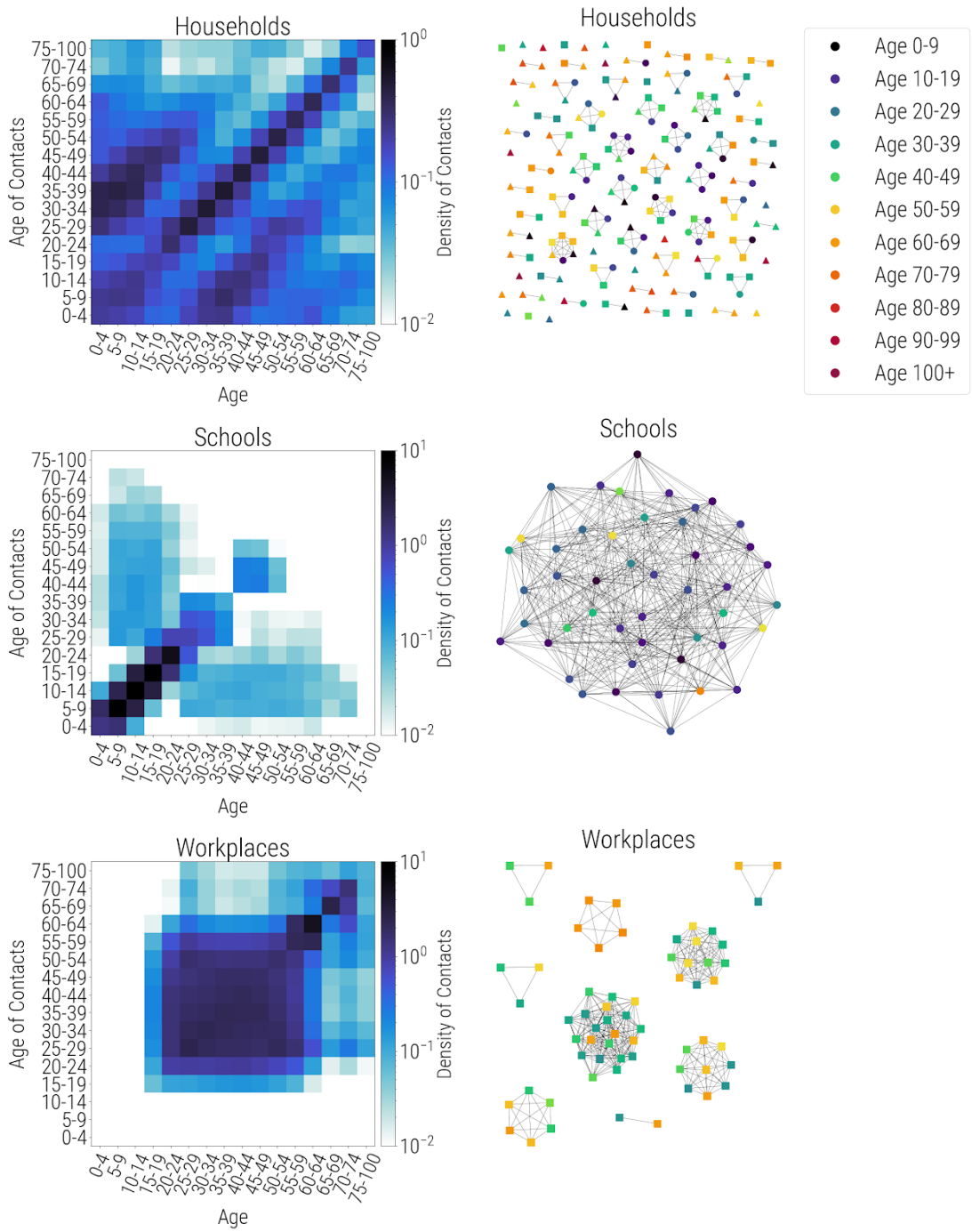


Fig. 4: Synthetic population networks for households (top), schools (middle), and workplaces (bottom). Age-specific contact matrices are shown on the left, while actual connectivity patterns for a 127-person subsample of a population of 10,000 individuals are shown on the right. All individuals are present in the household network, including some with no household connections. A subset of these individuals, including teachers, are present in the school network (circles); another subset is present in workplace networks (squares); some individuals are in neither school nor work networks (triangles).

2.4.2.1 Households

SynthPops generate individuals within households using data on the distribution of ages, household sizes, and the age of reference individuals per household for a given population. The algorithm first generates household sizes from the household size distribution, and then assigns a reference individual (for example, the head of the household) with their age sampled conditional on the household size. To construct the other household members, location-specific household age mixing contact matrices and the population age distribution are used to infer the likely ages of household contacts for the reference person. Each column *c* of the contact matrix is treated as an age distribution of the household contacts for a person in the age group *c*. The ages of other household members are then sampled conditional on the age of the reference person for the household.

2.4.2.2 Schools

A similar approach is used to construct schools. School enrollment data, available from census studies or survey data can be used to inform enrollment rates by age, school sizes, and student-teacher ratios. The SynthPops algorithm first chooses a reference student for the school conditional on enrollment rates to infer the school type, and then uses the age mixing contact matrix in the school setting to infer the likely ages of the other students in the school. Students are drawn from an ordered list of households, such that they reproduce an approximation of the neighborhood dynamics of children attending area schools together. Teachers and other non-teaching staff (e.g., administrative or cleaning staff) are drawn from the adult population comprising the labor force and assigned to schools as needed, reflecting average student-teacher and student-staff ratio data. With large schools, it is unlikely for each student, teacher, or other staff member to be in close contact with all other individuals. Instead, for each individual in the school layer we model their close and effective contacts as a subset of contacts from their school who can infect them by sampling a random set of *n* other individuals in their school, where *n* is drawn from a Poisson distribution with rate parameter λ_s equal to the average class size (λ_s = 20 as a default).

2.4.2.3 Workplaces and community

The labor force is drawn using employment rates by age, and non-teachers are assigned to workplaces using data on establishment sizes. Workers are assigned to workplaces using a similar method with an initial reference worker sampled from the labor force and their co-workers inferred from age mixing patterns within the workforce. All workers (teachers included) are drawn at random from the population, to reflect the general mixing of adults from different neighborhoods at work. Similar to the school layer, large workplaces are unlikely to be fully connected graphs of contacts. Instead, for each worker, we model their close contacts as a subset of *n* contacts from other individuals in their workplace, where *n* is drawn from a Poisson distribution with rate parameter λ_w equal to the estimated maximum number of close contacts in the workplace (λ_w = 20 as a default).

For contacts in the general community, we draw *n* random contacts for each individual from other individuals in the population, where *n* is drawn from a Poisson distribution with rate parameter λ_c equal to the expected number of contacts in the general community (with λ_c = 20 as a default, as above). Connections in this layer reflect the nature of contacts in shared public spaces like parks and recreational spaces, shopping centers, community centers, and public transportation. All links between individuals are considered undirected to reflect the ability of either individual in the pair to infect each other.

The generated multilayer network of household, school, work, and community network layers presents a population with realistic microstructure. This framework can also be extended to consider more detailed interactions in key additional settings, such as hospitals, encampments, shelters for those experiencing homelessness, and long term care facilities.

2.4.3 Hybrid networks

Covasim contains a third option for generating contact networks, which captures some of the realism of the SynthPops approach but does not require as much input data, and is more readily adaptable to other settings. As such, it is a "hybrid" approach between a fully random network and a fully data-derived network. As with SynthPops, each person in the population has contacts in their household, school (for children), workplace (for adults), and community. A population of individuals is generated according to a location-specific age distribution, and each individual is randomly assigned to a household using location-specific data on household sizes (using the pre-loaded UN data described above).

Unlike SynthPops, the hybrid algorithm does not account for the distribution of ages within a household. Children are assigned to schools and adults to workplaces, each with a user-specified number of fixed daily contacts (by default, Poisson-distributed with means of 20 for schools and 16 for workplaces, chosen to match the mean values for SynthPops networks). Individuals additionally have contacts with others in the community (by default, Poisson distributed with a mean of 20). All children and young adults aged between 6 and 22 are assigned to schools and universities, and all adults between 22 and 65 are assigned to workplaces. This distinguishes it from SynthPops where enrollment or employment varies depending on the given data. A comparison of the different population structure options available in Covasim is listed in Table 3.

Table 3: Comparison of population options in Covasim.

Population type	**Data requirements**	**Best suited for**	**Not well suited for**
Random networks	None	Models of transmission in special settings such as prisons or cruise ships	Large or complex populations
Hybrid networks	Data on the age distribution and household sizes for each country are pre-loaded No additional data is required, but users can optionally specify the daily number of school, workplace, and community contacts	Population network models in data-rich settings; adaptable and suited to most modeling contexts	Populations with high heterogeneity in contact patterns or size distributions
SynthPops networks	Household, school, workplace, and community age mixing patterns School size distributions, enrollment rates by age, student-teacher ratios Workplace size distributions, employment rates by age Number of households, size distribution, and age/sex distribution	Complex populations in data-rich settings	Settings where the data requirements cannot be met, or where other social settings are critical contexts for disease transmission

2.5 Interventions

A core function of Covasim is modeling the effect of interventions on disease transmission or health outcomes, to understand the impact that different policy options may have in a specific setting. In general, interventions are modeled as changes to parameter values. Covasim has built-in implementations of the common interventions described below, as well as the ability for users to create their own interventions, which can either be derived from the base intervention class, or be simple functions that modify the simulation object. Both built-in and user-defined interventions have full access to the simulation object at each timestep, which means that user-defined interventions can dynamically modify any aspect of the simulation. This can be used to create interventions more specific than those included by default in Covasim, such as age-specific physical distancing or quarantine, or interventions that are dynamically "triggered" based on the current or past state of the simulation.

2.5.1 Physical distancing, masks, and hygiene

The most basic intervention in Covasim is to reduce transmissibility (β) starting on a given day. This can be used to reflect both (a) reductions in transmissibility per contact, such as through mask wearing, personal protective equipment, hand-washing, and maintaining physical distance; and (b) reductions in the number of contacts at home, school, work, or in the community. However, Covasim also includes an "edge-clipping" intervention (considering a contact between two agents as a weighted "edge" between two "nodes"), where β remains unchanged but the number of contacts that person has is reduced. Complete school and workplace closures, for example, can be modeled either by setting β to 0, or by removing all edges in those contact layers; partial closures can be modeled by smaller reductions in either β or the number of contacts.

In general, both types of interventions have similar impact – for example, halving the number of contacts and keeping β constant will produce very similar epidemic trajectories as halving β and keeping the number of contacts constant. However, the distinction becomes important when considering the interaction between physical distancing and other interventions. For example, in a contact tracing scenario, the number of contacts who require tracing, number of tests performed, and number of people placed in quarantine are all strongly affected by whether physical distancing is implemented as a reduction in β of specific edges, or removing those edges entirely.

2.5.2 Testing and diagnosis

Testing can be modeled in two different ways within Covasim, depending on the format of testing data and purpose of the analysis. The first method allows the user to specify the probabilities that people with different risk factors and levels of symptoms will receive a test on each day. Separate daily testing probabilities can be entered for those with/without symptoms, and those in/out of quarantine. The model will then estimate the number of tests performed on each day. The second method allows the user to enter the number of tests performed on each day directly, including multipliers on the probability of a person receiving a test if they have symptoms, are in quarantine, or are over a certain age. This method will then allocate the tests among the population. If data on the number of daily tests performed each day are available, the second method is preferable.

Once a person is tested, the model contains a delay parameter that indicates how long people need to wait for their results, as well as a loss-to-follow-up parameter that indicates the probability that people will not receive their results. Additional parameters control the sensitivity and specificity of the tests.

2.5.3 Contact tracing

Contact tracing corresponds to notifying individuals that they have had contact with a confirmed case, so that they can be quarantined, tested, or otherwise change their behavior. Contact tracing in Covasim is parameterized by the probability that a contact can be traced, and by the time taken to identify and notify contacts. Both parameters can vary by type of contact, and can be controlled by the user. For example, it may be reasonable to assume that people can trace members of their household immediately and with 100% probability, while tracing work colleagues may take several days and may be incomplete. Digital contact tracing can be approximated in Covasim as a standard contact-tracing intervention with zero delays, with the caveat that tracing multiple steps (i.e., contacts of contacts) within a single day would require a custom intervention.

2.5.4 Isolation of positives and contact quarantine

Isolation (referring to behavior changes after a person is diagnosed with COVID-19) and quarantine (referring to behavior changes after a person is identified as a known contact of someone with confirmed or suspected COVID-19) are the primary means by which testing interventions reduce transmission. In Covasim, people diagnosed with COVID-19 can be isolated. Their contacts who have been traced can be placed in quarantine with a specified level of compliance; people in quarantine may also have an increased probability of being tested. People in isolation or quarantine typically have a lower probability of infecting others (if infectious) or of acquiring COVID-19 (if quarantined and susceptible). The default reductions for isolation are 70% in the household and 90% in school, work, and community layers, while quarantine is assumed to have lower compliance (40% reduction in the household and 80% in other layers). However, if psychosocial support is not provided to people in home isolation or quarantine, there may be an increased risk of passing on infection to, or acquiring infection from, other household members. For performance reasons, isolation and quarantine are implemented as reductions in per-contact transmission risk rather than changes in the number of contacts; for realistic parameter values (i.e., $\beta \ll 1$), the difference between these implementations should be negligible.

2.5.5 Vaccines and treatments

Pharmaceutical interventions, especially vaccines, are an increasingly important part of public health responses to COVID-19. However, there are significant modeling challenges due to the large number of vaccine candidates under investigation, coupled with the considerable uncertainty regarding their properties – such as the extent to which they block acquisition and transmission as well as symptoms, how much protection is conferred by a single dose, the extent to which immunity wanes over time, and their effectiveness against different COVID-19 strains (70). Vaccines in Covasim are modeled by adjusting individuals' susceptibility to infection and probability of developing symptoms after being infected; both of these modifications affect the overall probability of progressing to severe disease and death. Additional flexibility, including waning efficacy and differential effectiveness across variants, will be incorporated as trial results become available. Though treatments for COVID-19 have so far had only modest results in clinical trials (71), they can be implemented in Covasim as interventions that reduce the probability of progressing to severe disease or death.

2.6 Additional features

2.6.1 Data inputs

In addition to the demographic and contact network data available via SynthPops, Covasim includes interfaces to automatically load COVID-19 epidemiology data, such as time series data on deaths and diagnosed cases, from several publicly available databases. These databases include the Corona Data Scraper (coronadatascraper.com), the European Centre for Disease Prevention and Control (ecdc.europa.eu), and the COVID Tracking Project (covidtracking.com). At the time of writing, these data are available for over 4,000 unique locations, including most countries in the world (administrative level 0), all US states and many administrative level 1 (i.e., subnational) regions in Europe, and some administrative level 2 regions in Europe and the US (i.e., US counties).

2.6.2 Dynamic rescaling

One of the major challenges with agent-based models is simulating a sufficient number of agents to capture an epidemic at early, middle, and late stages, without requiring cumbersome levels of memory or processor usage. Whereas compartmental SEIR models require the same amount of computation time regardless of the population size being modeled, the performance of agent-based models typically scales linearly or supralinearly with population size (see Section 2.7.1). As a consequence, many agent-based models, including Covasim, include an optional "scaling factor", where a single agent in the model is assumed to represent multiple people in the real world. A scaling factor of 10, for example, corresponds to the assumption that the epidemic dynamics in a city of 2 million people can be considered as the sum of the epidemic dynamics of 10 identical subregions of 200,000 people each.

However, the limitation of this approach is that it introduces a discretization of results: model outputs can only be produced in increments of the scaling factor, so relatively rare events, such as deaths, may not have sufficient granularity to reflect the epidemic behavior at a small scale. In addition, using too few agents in the model introduces stochastic variability patterns that do not reflect real-world processes in the entire population.

To circumvent this, Covasim includes an option for dynamic rescaling. Initially, when the epidemic is small, there is no scaling performed: one agent corresponds to one person. Once a certain threshold is reached, however (by default, 5% of the population is non-susceptible), the non-susceptible agents in the model are downsampled and a corresponding scaling factor is introduced (by default, a factor of 1.2 is used). For example, in a simulation of 100,000 agents representing a true population of 1 million with a threshold of 10% and a rescaling factor of 2, dynamic rescaling would be triggered when cumulative infections surpass 10,000, leaving 90,000 susceptible agents; dynamic rescaling would then resample the non-susceptible population to 5,000 (now representing 10,000 people) and increase the number of susceptible agents to 95,000 (now representing 190,000 people), i.e. with every agent now counting as two. If the epidemic expands further, this process will repeat iteratively until the scale factor reaches its upper limit (which in this example is 10, and which would be reached after 100,000 cumulative infections). Through this process, arbitrarily large populations can be modeled, even starting from a single infection, maintaining a constant level of precision and computation time throughout.

2.6.3 Health system capacity

Individuals in the model who have severe and critical symptoms are assumed to require regular and intensive care unit (ICU) hospital beds, respectively, including ventilation in the latter case. The number of available hospital beds (ICU and otherwise) beds are input parameters. If the model estimates that the number of severe/critical cases is greater than the number of available non-ICU/ICU beds, then the health system capacity is exceeded. This means that severely ill individuals have an increased probability of progressing to critical, and critically ill individuals who are unable to access treatment have an increased mortality rate (by default, both by a factor of 2).

2.6.4 Importations

The spatial movement of agents is not currently modeled explicitly in Covasim, and the population size for a given simulation is fixed. Thus, importations are implemented as spontaneous infections among the susceptible population. This corresponds to agents who become infected elsewhere and then return to the population.

2.6.5 Model outputs

By default, Covasim outputs three main types of result: "stocks" (e.g., the number of people with currently active infections on a given day), "flows" (e.g., the number of new infections on a given day), and "cumulative flows" (e.g., the cumulative number of infections up to a given day). For states that cannot be transitioned out of (e.g. death, plus recovery if reinfection is not considered), the stock is equal to the cumulative flow. Flows that are calculated in the model include: the number of new infections and the number of people who become infectious on that timestep; the number of tests performed, new positive diagnoses, and number of people placed in quarantine; the number of people who develop mild, severe, and critical symptoms; and the number of people who recover or die. The date of each transition (e.g., from critically ill to dead) is also recorded. By default, these results are summed over the entire population on each day; results for subpopulations can be obtained by defining custom analyzers, as described in Section 2.6.7.

2.6.6 Reproduction number and doubling time

In addition to these core outputs, Covasim includes several outputs for additional analysis. For example, several methods are implemented to compute the effective reproduction number R_e. Numerous definitions of R_e exist; in standard SIR modeling, the most common definition ("method 1") is (72):

$$R_e = R_0 S/N$$

where R_0 is the basic reproduction number, S is the number of susceptibles, and N is the total population size. However, with respect to COVID-19, many authors instead define R_e to include the effects of interventions, due to the implications that $R_e = 1$ has for epidemic control.

A second common definition of R_e ("method 2") is to first determine the total number of people who became infectious on day t, then count the total number of people these people went on to infect, and then divide the latter by the former. "Method 3" is the same as method 2, except it counts the number of people who stopped being infectious on day t (i.e., recovered or died), and then counts the number of those people infected. Unlike in a compartmental model, where R_e can only be estimated by using

simplifying assumptions, in an agent-based model, methods 2 and 3 can be implemented by simply counting exactly how many secondary infections are caused by each primary infection. By doing so, all details of the epidemic – including time-varying viral loads, population-level and localized immunity, interventions, network factors, and other effects – are automatically incorporated, and do not need to be considered separately.

While methods 2 and 3 are implemented in Covasim, they have the disadvantage that they introduce significant temporal blurring, due to the potentially long infectious period (and, for method 3, the long recovery period). To avoid this limitation, the default method Covasim uses for computing R_e is to divide the number of new infections on day *t* by the number of actively infectious people on day *t*, multiplied by the average duration of infectiousness ("method 4"). This definition of R_e is nearly identical to the definition of the "instantaneous reproductive number" in Gostic et al. (73), which in that study is used as the ground truth against which other R_e estimators are compared.

Covasim also includes an estimate of the epidemic doubling time, computed similarly to the "rule of 69.3" (74), specifically:

$$T = \frac{w \log(2)}{\log(n_i(t)/n_i(t-w))}$$

where *T* is the doubling time, *w* is the window length over which to compute the doubling time (3 days by default), and $n_i(t)$ is the cumulative number of infections at time *t*.

2.6.7 Analyzers

In addition to interventions, Covasim also includes a library of "analyzers". Like interventions, in principle they can access and modify any aspect of the simulation state. However, they are typically used to record additional details about the internal state of the model that are not included as standard outputs (e.g., the age distribution of infections at a given point in time). By convention, interventions and analyzers differ in that interventions modify the state of the simulation (and are applied at the beginning of each timestep), while analyzers record the state (and are applied at the end of each timestep).

2.6.8 Calibration

The process of calibration involves finding parameter values that minimize a function that measures the difference between observed data (which typically includes daily confirmed cases, hospitalizations, deaths, and number of tests conducted) and the model predictions. Since most data being calibrated to are time series count data, this function is defined as:

$$L = \sum_s \sum_t w_s f(c_d^{s,t}, c_m^{s,t})$$

where s refers to the type of data observed (such cumulative confirmed cases or number of deaths); t is the time index; w_s is the weight associated with s; $c_d^{s,t}$ and $c_m^{s,t}$ are the counts from the data and model, respectively, for this time series at this time index; and *f* is the loss, objective, or goodness-of-fit function (e.g., normalized absolute error, mean absolute error, mean squared error, or the Poisson test statistic (76)). By default, Covasim calculates the loss using normalized absolute error. Depending on underlying distributional assumptions, minimizing the normalized absolute error can sometimes give

parameter estimates that are equivalent to the estimates that maximize the log-likelihood (or an approximation thereof, as in approximate Bayesian computation (74)). Intuitively, most distributional assumptions mean that larger errors imply a lower log-likelihood. However, we do not make explicit distributional assumptions, so caution is advised with treating them as statistically rigorous likelihoods.

Calibrating any model to the COVID-19 epidemic is an inherently difficult task: not only is there significant uncertainty around the reported data, but there are also many possible combinations of parameter values that could give rise to these data. Thus, in a typical calibration workflow, most parameters are fixed at the best available values from the literature, and only essential parameters (for example, β) are allowed to vary.

Currently, calibration must be performed externally to Covasim. However, since a single model run returns a scalar loss value, these runs can be easily integrated into standardized calibration frameworks. Any standard optimization library – such as the optimization module of SciPy – can be easily adapted (as long as it can handle stochastic results, which standard gradient descent cannot), as can more advanced methods such as the adaptive stochastic descend method of the Sciris library (77), or Bayesian approaches such as history matching (78) and sequential Monte Carlo methods (79). To date, the Optuna hyperparameter optimization library (80) has proven to be the most effective approach for calibration, and an example implementation is included in the codebase.

2.7 Software architecture

Covasim was developed for Python 3.8 using the SciPy (scipy.org) ecosystem (81). It uses NumPy (numpy.org), Pandas (pandas.pydata.org), and Numba (numba.pydata.org) for fast numerical computing; Matplotlib (matplotlib.org) and Plotly (plotly.com) for plotting; and Sciris (sciris.org) for data structures, parallelization, and other utilities.

The source code for Covasim is available via both the Python Package Index (via `pip install covasim`) and GitHub (github.com/institutefordiseasemodeling/covasim). Covasim is fully open-source, released under the Creative Commons Attribution-ShareAlike 4.0 International Public License. More information is available at covasim.org, with full documentation and a comprehensive set of tutorials available at docs.covasim.org.

2.7.1 Performance

All core numerical algorithms in the Covasim integration loop – specifically, calculating intra-host viral load, per-person susceptibility and transmissibility, and which contacts of an infected person become infected themselves – are implemented as highly optimized 32-bit array operations in Numba. For further efficiency, agents are not represented as individual objects, but rather as indices of one-dimensional state arrays (Fig. 5). This avoids the need to use an explicit for-loop over each agent on every integration timestep, increasing performance by more than an order of magnitude. Similarly, contacts between all agents in the model are stored as a single array of "edges" per contact layer.

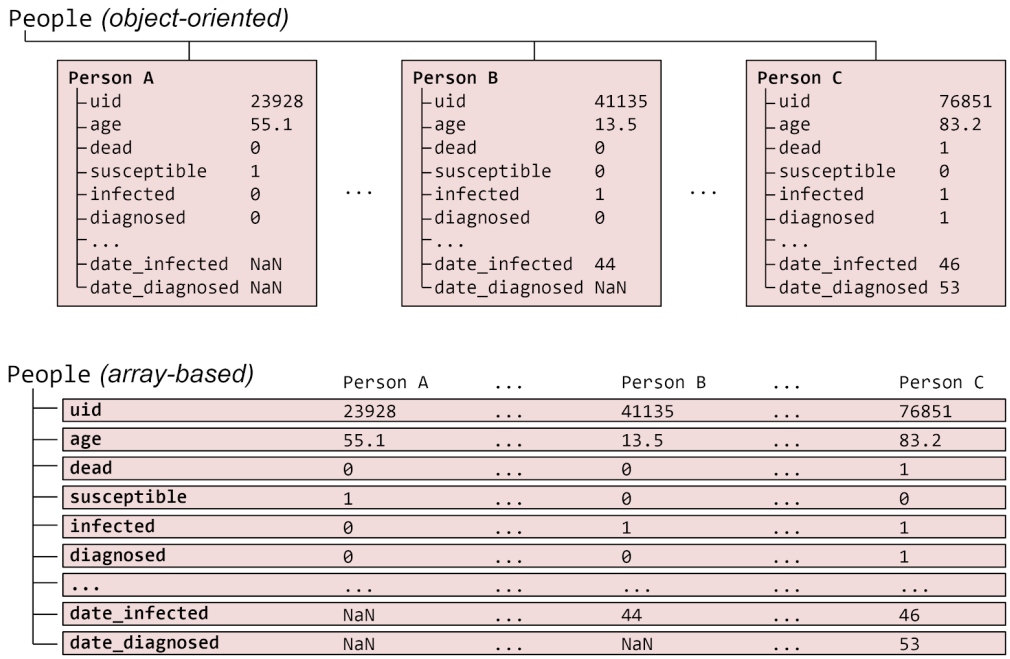


Fig. 5: Illustration of the standard object-oriented approach for implementing agent-based models (top), where each agent is a separate object, compared with the approach used in Covasim (bottom), where agents are represented as slices through a set of state arrays. Dots (...) represent omitted entries. In practice, each agent has 40 states, and there are typically hundreds of thousands of agents.

As shown in Fig. 6, these software optimizations allow Covasim to achieve high levels of performance, despite being implemented purely in Python. Performance scales linearly with population size over multiple orders of magnitude: memory scales at a rate of roughly one agent per 1 KB of memory, while single-core compute time (benchmarked on an Intel i9-8950HK laptop processor) scales at a rate of roughly 7 million simulated person-days per second of CPU time. These speed and memory use results are comparable to OpenABM-Covid19, despite the latter being implemented in C (23). One consequence of the array-based implementation is that compute time depends on the number of agents and the number of connections per agent, but is independent of the number of infected agents; this is because uninfected agents are simply represented as zeros in the transmission probabilities vector.

Due to Covasim's computational efficiency, it is feasible to run realistic scenarios, such as tens of thousands of infections among a susceptible population of hundreds of thousands of people for a duration of 12 months, in under a minute on a personal laptop. Covasim is also suited to high-performance computing environments, with support for parallelization via the built-in "multiprocessing" library; it can also be adapted easily to other parallel processing libraries such as Celery and Dask. Although in some special situations it is possible to split a single simulation across multiple cores, parallel processing is used primarily to run multiple independent simulations simultaneously, such as for uncertainty analyses or calibration.

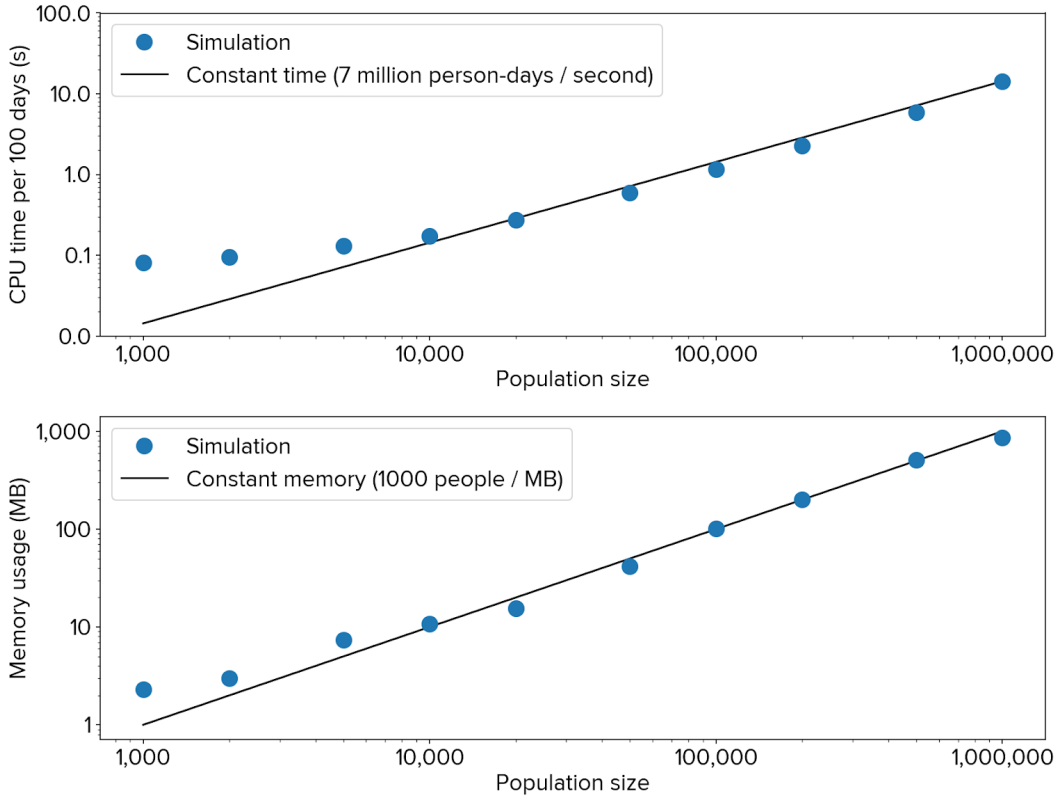


Fig. 6: Covasim performance in terms of processor usage (top) and memory usage (bottom), for the number of agents shown, simulated for 100 days. There is roughly linear scaling over three orders of magnitude of population size.

2.7.1 Deployment and access

While Covasim is primarily intended to be used via Python scripts, a number of other options for using it are also available. A simple webapp for Covasim has been developed, based on Vue.js (for the frontend), ScirisWeb (for communicating between the frontend and the backend), Flask (for running the backend), and Gunicorn/NGINX (for running the server); this webapp is available at app.covasim.org. A screenshot of the user interface is shown in Fig. 7. A pre-built version of Covasim, including the webapp, is also available on Docker Hub (hub.docker.com). Covasim can also be run via R using the "reticulate" library, and from the command line via the "fire" library.

2.7.2 Software tests

Covasim includes an extensive suite of both integration tests and unit tests; code coverage for version 2.1.1 is 89%. In addition, outputs from the default simulations for each version are compared against cached values in the repository; since random seeds are stored, results are exactly reproducible despite the stochasticity in the model. When new data become available and parameter values are updated, previous parameters are stored, ensuring that any changes affecting the model outputs are intentional, and that previous versions can be easily retrieved and compared against. The test suite includes unit tests (e.g., checking that sampling functions produce the specified distributions; that simulations loaded from file exactly match the original), functional tests (e.g., that a simulation run with a particular analyzer produces a plot), and end-to-end "scientific" tests (e.g., that an increase in mortality rate leads to more deaths, while adding NPIs leads to fewer).

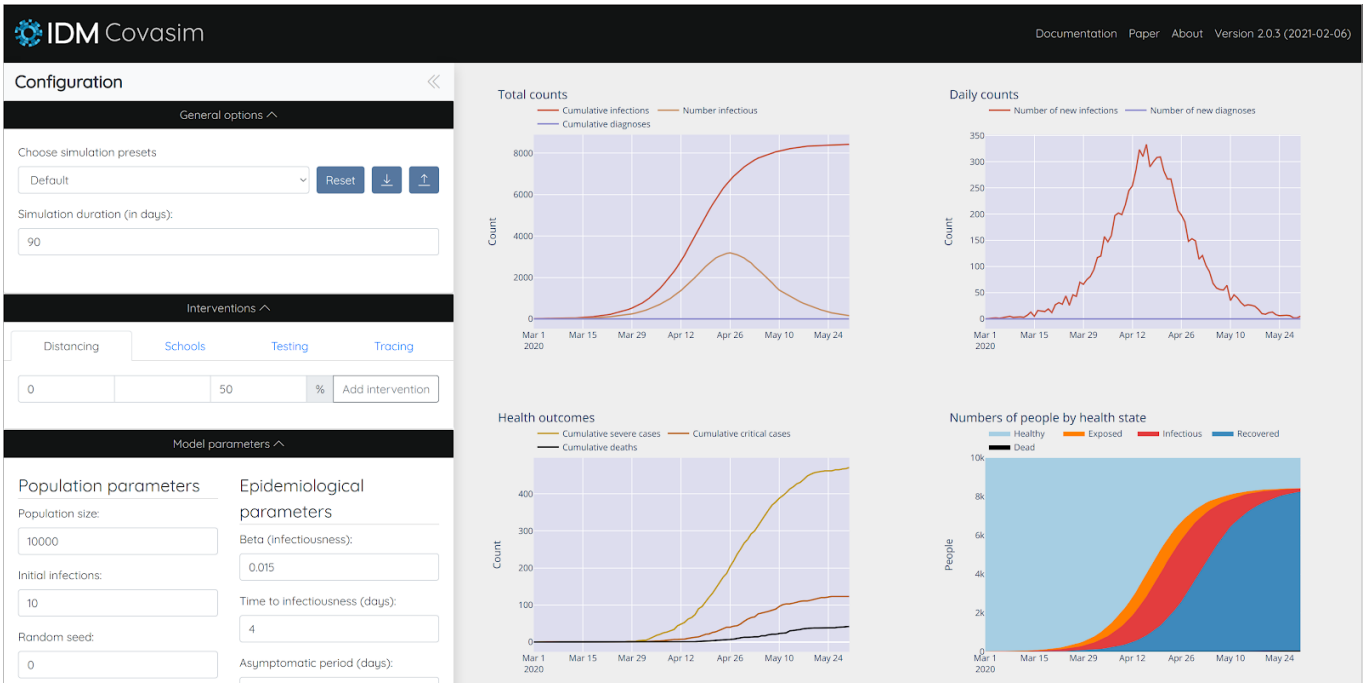


Fig. 7: Covasim webapp user interface; screenshot taken from http://app.covasim.org.

3 Results

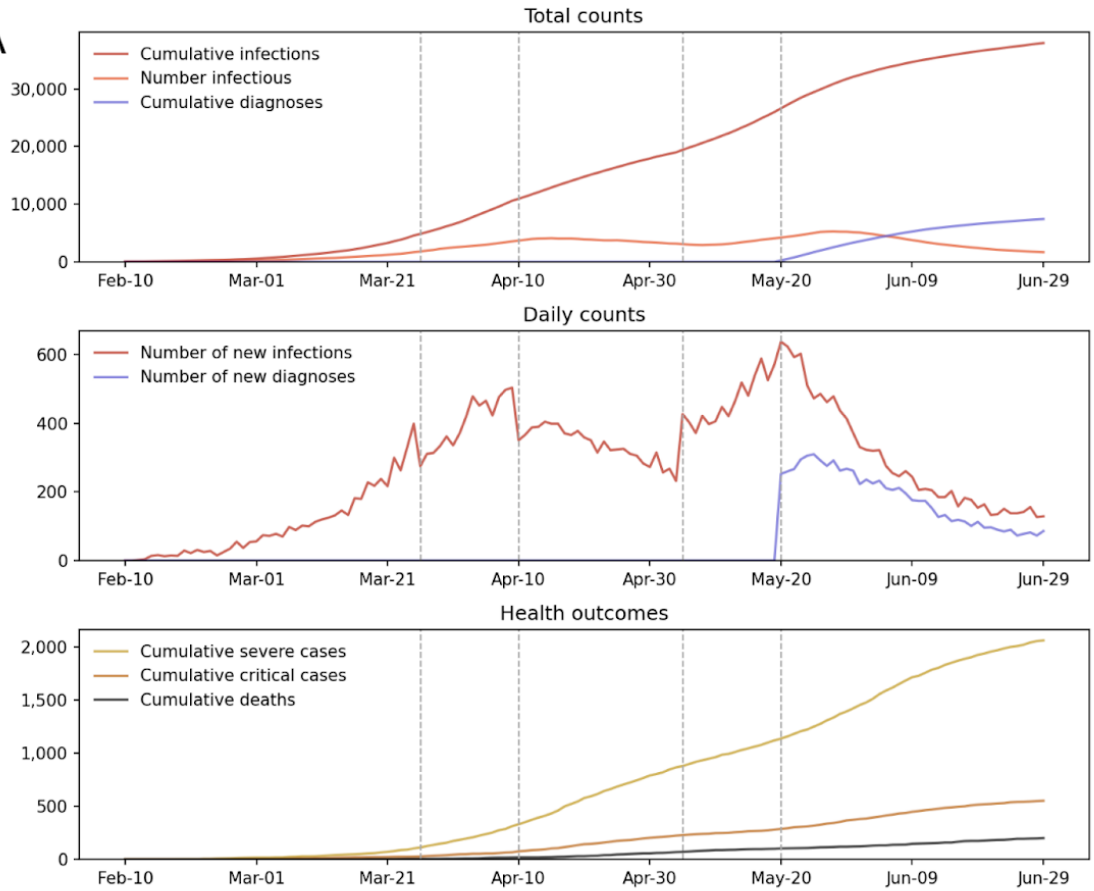
3.1 Example usage

Several of Covasim's standard features are illustrated in Fig. 8A. It represents a simulation of 200,000 people, from February 10 to June 29 2020, starting with 75 seed infections. After an initial 45 days of uncontrolled epidemic spread, the following interventions are applied: March 26, close schools and reduce work and community contacts to 70% of their original values; April 10, reduce work and community to 30% of their original values; May 5, reopen work and community to 80% of their original values; May 20, begin testing 10% of people with COVID-like illness each day, and trace the contacts of people who test positive.

By default, Covasim shows time series for key cumulative counts, daily counts, and health outcomes (including deaths). All plotting outputs are configurable, and results can also be saved in Excel, JSON, or NumPy formats for further processing. While a full Covasim application would likely include additional complexity regarding calibration and plotting, other aspects of the example shown in Fig. 8A are comparable to a real-world exploratory policy analysis. Despite this, the Python script used to generate Fig. 8A is only 28 lines; this code is listed in Fig. 8B.

In addition to running single simulations, Covasim also allows the user to run multiple simulations, which can be averaged over to determine forecast intervals. By default, the "80% forecast interval" is used, i.e. between the 10th and 90th percentiles of the simulated trajectories. Since these forecast intervals are typically produced by a combination of both stochastic variability ("aleatory uncertainty") and imperfect knowledge of the "true" parameter values ("epistemic uncertainty"), they should not be interpreted as statistically rigorous Bayesian credible intervals (82,83). Multiple simulations can also be used to quickly run different scenarios in parallel and compare them, as shown in Fig. 9.

A

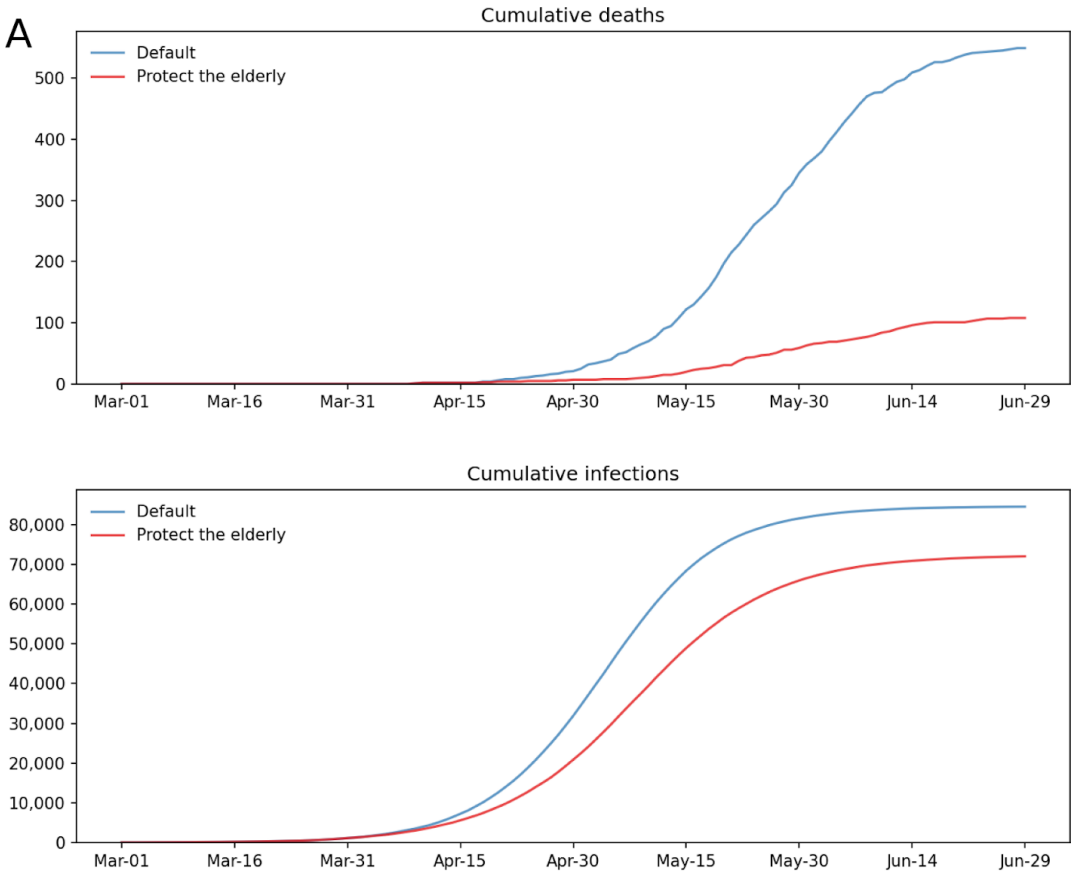


B

```
1  import covasim as cv
2
3  # Define the simulation parameters
4  pars = dict(
5      pop_size     = 200e3,        # Define a population size of 200,000 people
6      pop_infected = 75,           # Start with 75 infected individuals
7      beta         = 0.012,        # Calibrate overall transmission to this setting
8      pop_type     = 'hybrid',     # Use realistic household, school, and work contacts
9      start_day    = '2020-02-10', # First day of the simulation
10     end_day      = '2020-06-29', # Last day of the simulation
11 )
12
13 # Define the interventions
14 tr_probs = dict(h=0.9, s=0.7, w=0.7, c=0.3) # Probability that a contact in each layer will be traced
15 tr_time  = dict(h=0.0, s=1.0, w=1.0, c=3.0) # Time required to trace contacts in each layer (days)
16 m_days    = ['2020-03-26', '2020-04-10', '2020-05-05'] # Days on which mobility changes
17 m_changes = [0.7, 0.4, 0.8] # Define the mobility changes on these days
18 interventions = [
19     cv.clip_edges(days='2020-03-26', changes=0.0, layers='s'),        # Close schools
20     cv.clip_edges(days=m_days, changes=m_changes, layers=['w', 'c']), # Close/reopen work + community
21     cv.test_prob(start_day='2020-05-20', symp_prob=0.1, symp_quar_prob=0.8, test_delay=1), # Testing
22     cv.contact_tracing(start_day='2020-05-20', trace_probs=tr_probs, trace_time=tr_time)   # Contact tracing
23 ]
24
25 # Create and run the simulation
26 sim = cv.Sim(pars=pars, interventions=interventions)
27 sim.run()
28 sim.plot()
```

Fig. 8: (A) Illustrative example of a single run of a Covasim simulation. Interventions (described in the text) are shown as dashed vertical lines. (B) Full listing of the code for this simulation, including defining the parameters of the simulation (lines 4–11); defining the interventions (lines 14–23); and creating, running, and plotting the simulation (lines 26–28).

A



B

```
1  import covasim as cv
2
3  def protect_elderly(sim):
4      if sim.t == sim.day('2021-04-01'):
5          elderly = sim.people.age>70
6          sim.people.rel_sus[elderly] = 0.0
7
8  pars = {'pop_size':100e3, 'start_day':'2021-03-01', 'n_days':120}
9  s1 = cv.Sim(pars, label='Default')
10 s2 = cv.Sim(pars, label='Protect the elderly', interventions=protect_elderly)
11 cv.MultiSim([s1, s2]).run().plot(to_plot=['cum_deaths', 'cum_infections'])
```

Fig. 9: (A) Illustrative example of a scenario comparison using a simple custom intervention ("protecting the elderly", i.e. removing all transmission among people over age 70 after a certain date). (B) Full listing of the code for this simulation, showing the intervention definition (lines 3–6), and a compact way of creating the simulations, running them in parallel, and plotting them (lines 8–11).

3.2 Transmission analyses

The preceding examples illustrate some aspects of Covasim's core functionality that are used in most applications. More in-depth analyses are also possible, leveraging either the default outputs, or the fact that the full state of the model is accessible to the user at every timestep via custom analysis functions.

For example, detailed information about the transmission tree is stored for each simulation. This information can be used to determine the detailed microstructure of the infection patterns in a given simulation. Complete transmission trees for a small network under three different intervention scenarios are shown in Fig. 10, visualized via the ETE Toolkit (84). For realistically sized networks, it is not feasible to visualize entire transmission trees. However, their statistical properties can be analyzed to determine

transmission routes and potential intervention targets. For example, such information can be used to determine the net contribution of schools (or even teachers at schools) to the overall epidemic trajectory (27).

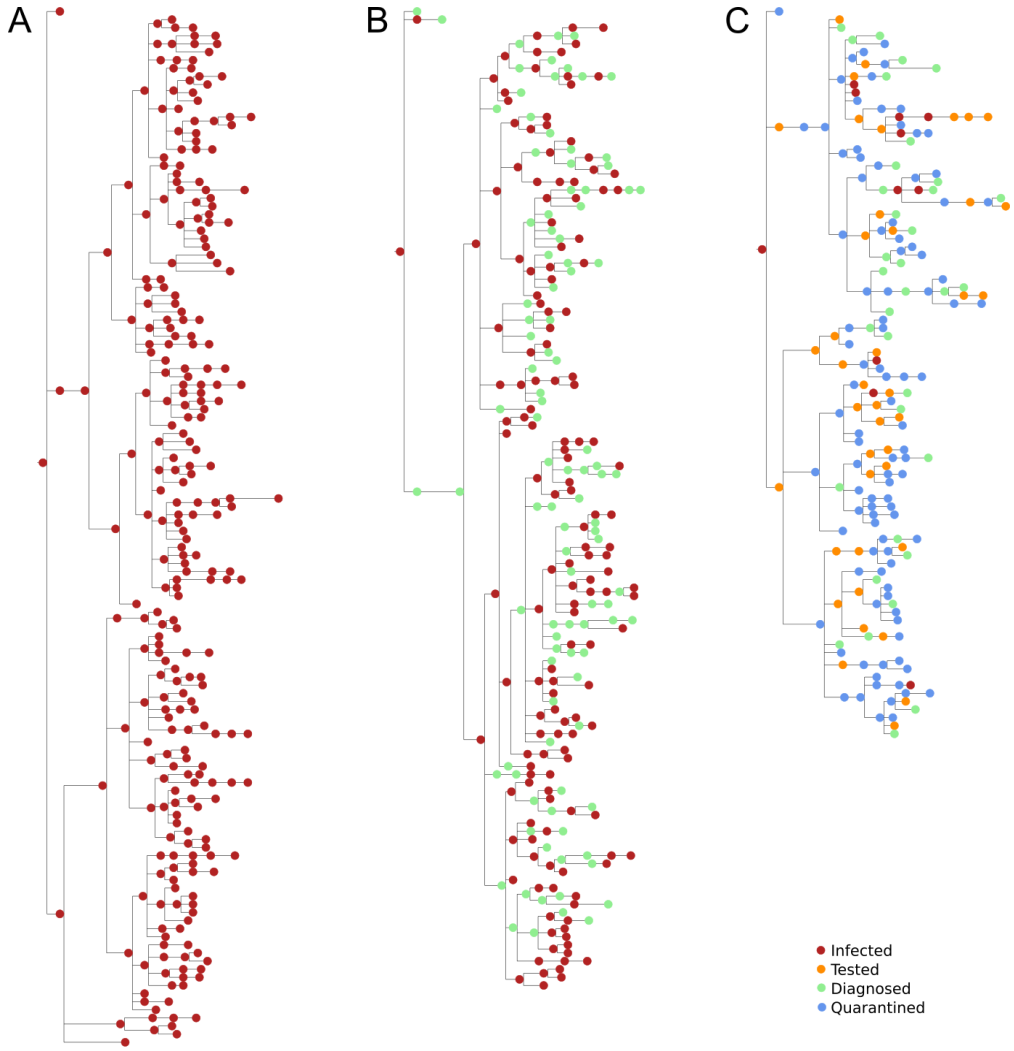


Fig. 10: Example transmission trees for a hypothetical population of 300 individuals with a single seed infection on day 1, with (A) no interventions, (B) testing only, and (C) testing plus contact tracing. Time is shown on the horizontal axis, with each tree representing approximately 90 days. The vertical size of each tree is proportional to the total number of infections.

3.3 Case study

Here we provide a case study of how Covasim was used to inform a policy decision in King County (the local government area that includes the city of Seattle), Washington, USA; a full description of the methodology used is given in (25). Briefly, we used Optuna to calibrate Covasim to epidemiological and program data from January 27 to November 14 2020; these data are available from the Public Health Seattle King County data dashboard (3). We then ran the model with eight different calibrated parameter sets (with multiple parameter sets used to capture parametric uncertainty) to (a) estimate unobserved quantities, such as the number of new infections and the case detection rate; (b) estimate the impact of proposed new mobility restrictions (such as limiting indoor dining) scheduled to start on November 16, which we estimated would result in a 15% reduction in transmission (85); and (c) compare this scenario with counterfactual scenarios of either not implementing the scheduled restrictions, or by implementing them together with increased testing and contact tracing.

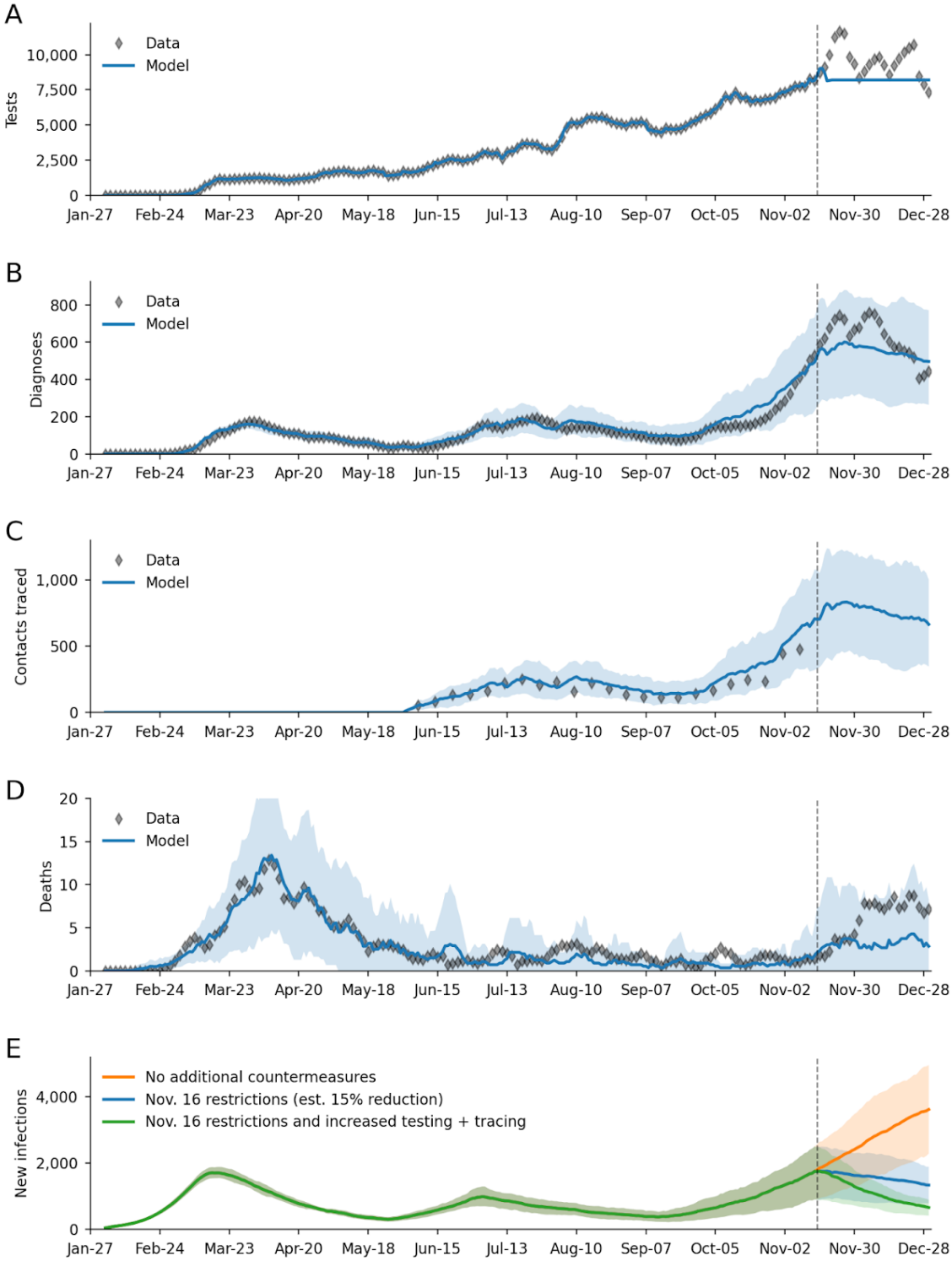


Fig. 11: Example calibration of Covasim to data from Seattle/King County, Washington, USA from January 27 to November 14 2020 (dashed line), with projections until December 31, including additional restrictions imposed on November 16. (A) Number of daily COVID-19 tests, which are used as input data. (B) Calibration to the number of daily COVID-19 diagnoses. (C) Calibration to the number of daily contacts traced (weekly averages shown; data past prediction date are not available). (D) Calibration to the number of daily COVID-19 deaths. (E) Projections of the number of new infections if restrictions had not been implemented, with the restrictions as implemented, and if restrictions were implemented together with increases in testing and contact tracing. Bands show 80% forecast intervals; data are rolling 7-day averages to account for weekend reporting delays.

As shown in Fig. 11, Covasim was able to capture numerous features of the epidemic during the calibration period, including numbers of tests and contacts traced (which were used as input data, along with mobility data from SafeGraph; see safegraph.com); the three infection "waves" (spring, summer, and fall); changes in test positivity rate (not shown), and numbers of deaths. During the scenario period, we assumed that the number of tests conducted per day would remain constant at the average value from the previous 7 days (Fig. 11A).

Despite a rapid increase of cases in the preceding weeks, the model predicted counterintuitively that even these modest mobility restrictions would be sufficient to stop the rise in cases (Fig. 11B), a projection that turned out to be accurate. (Note that using actual testing data for this period, rather than assuming a constant number of tests, would have resulted in an even more accurate prediction of diagnoses, though of course these data were not available at the time the prediction was made). While the model correctly predicted the trend in cases, it underestimated the number of deaths (Fig. 11C), although the observations were still within the 80% forecast interval (the large uncertainty interval for deaths is a consequence of the small numbers of events being predicted, i.e., fewer than 10 deaths per day; this forecast interval includes both parametric and stochastic uncertainty, as described in Section 3.1). This underestimate was likely due to assuming a continuation of infection patterns that occurred over the summer and early fall, during which younger adults were disproportionately infected compared to older ones.

Finally, we predicted that had the additional restrictions not been implemented, by the end of the year, daily infection rates would have been roughly three as high as actually occurred (Fig. 11D). Had testing and contact tracing programs been rapidly scaled up (by 50% and five fold respectively), we estimated the number of infections would have been approximately halved. These predictions helped provide quantitative support for public health decisions regarding mobility restrictions and increased testing.

4 Discussion

The COVID-19 pandemic has presented an unprecedented challenge to the disease modeling community in terms of requiring rapid, accurate predictions, often based on extremely limited data, with consequences of global scale. Covasim was developed to help policymakers make decisions based on the best available data, while taking into account the large uncertainties that remain in terms of COVID-19 transmission dynamics, disease progression, and other aspects of its biology, such as the proportions of asymptomatic and presymptomatic transmission.

We prioritized five different factors when developing Covasim: rapid development process, computational performance, flexibility, simplicity for users, and simplicity for developers. Striking a balance between these factors required making certain tradeoffs. For example, choosing to implement Covasim in Python instead of C++ or Java significantly reduced development time and increased simplicity for users and developers; however, it imposed a large penalty on performance. While we were able to solve this by using Numba and vectorized state arrays in place of object-oriented agents, this implementation increased development time and increased the complexity for developers. Another tradeoff we encountered is that while the gold standard in simplicity of use remains interactive webapps (86), the limited flexibility such webapps provide means that most Covasim users to date have instead used Python scripts to run analyses.

Beyond implementation tradeoffs, it is worth noting that in many cases, compartmental models offer

simpler, faster, and more robust results than agent-based models such as Covasim. Indeed, many of the most influential COVID-19 models that have been developed to date have been compartmental models (4,85,87,88). However, compartmental models have two major limitations. First, they cannot be easily adapted to changing epidemic conditions, such as new strains or multiple types of vaccine, since these often require a combinatorial explosion in the number of compartments (89,90). Second, they are unsuitable for answering questions that depend on details of behavior at the individual level, such as superspreading events, transmission within multigenerational households, school classroom cohorting, and contact tracing. While it is possible to approximate some of these phenomena in compartmental models (91,92), these approximations typically exclude important factors such as time delays. Some of the issues regarding compartmental models' predictive performance (93–95) may be partly a consequence of their inability to capture key mechanisms of epidemic spread. While agent-based models, including Covasim, are difficult to deploy widely enough, and calibrate quickly enough, to be a feasible replacement for compartmental models, they can provide a mechanistic understanding of the COVID-19 epidemic in ways that compartmental models cannot.

4.1 Limitations of Covasim

Covasim is subject to the usual limitations of mathematical models, most notably constraints around the degree of realism that can be captured. For example, human contact patterns are intractably complex, and the algorithms that Covasim uses to approximate these are necessarily quite simplified.

Like all models, the quality of the outputs depends on the quality of the inputs, and many of the parameters on which Covasim relies are still subject to large uncertainties. Most critically, the proportion of asymptomatics and their relative transmission intensity, and the proportion of presymptomatic transmission, strongly affect the number of tests required in order to achieve workable COVID-19 suppression via testing-based interventions.

Dynamical models are commonly validated by comparing their projections against data on what actually happened, as shown in the case study (Fig. 11). However, several challenges are commonly encountered when using this approach for COVID-19, including (a) data quality issues (such as low case detection rates and under-reporting of deaths); (b) the difficulty of predicting future social and political responses that would significantly impact model projections (such as the timing of school and workplace reopening, or a sudden increase in testing rates, as in the case study presented above); and (c) the fact that model-based projections themselves have the potential to influence policy decisions, e.g., optimistic model projections may lead to relaxed policies, which in turn will lead to worse outcomes than predicted, while pessimistic model projections may lead to stricter policies, which in turn will lead to better outcomes than predicted.

4.2 Future directions

More than a year after the emergence of SARS-CoV-2, our understanding of the pandemic is still evolving rapidly, especially regarding the risks posed by variant strains and the opportunities offered by vaccine candidates. These two issues currently present the most important questions regarding epidemic control, and hence are the two most active areas of Covasim development. Model parameter values are also continually updated as new data become available. Future development plans also include the incorporation of more detailed populations and networks, including healthcare workers, different types of industry, spatial mixing patterns, and the socioeconomic and racial disparities present in both transmission patterns and health outcomes. With the deployment of vaccines come additional questions

and interest regarding the lifting of mobility restrictions and social distancing guidelines, as well as questions about equitable vaccine distribution to different populations around the world. Additional Covasim development of data driven modeling of mobility between regions will help address the modeling and identification of the risk of importation to regions with fewer resources for early detection and treatment as pre-pandemic mobility gradually returns to parts of the world. Finally, we are committed to continuing our collaborations with stakeholders and policymakers around the globe, to work with them in determining how COVID-19 suppression can be achieved via a combination of distancing, testing, contact tracing, and vaccination.

Acknowledgements and contributions

Author contributions

Covasim model development was led by Cliff Kerr, Robyn Stuart, Romesh Abeysuriya, and Daniel Klein, with additional support by Gregory Hart, Katherine Rosenfeld, Prashanth Selvaraj, Rafael Núñez, Jamie Cohen, Lauren George, and Michał Jastrzębski. The SynthPops model was developed by Dina Mistry, with additional support by Cliff Kerr, Romesh Abeysuriya, Daniel Klein, and Lauren George. The health systems component was based on a model developed by Brittany Hagedorn. Analyses were performed by Cliff Kerr, Dina Mistry, Robyn Stuart, Katherine Rosenfeld, Gregory Hart, Rafael Núñez, Prashanth Selvaraj, and Jamie Cohen. Literature and code reviews were performed by Anna Palmer, Dominic Delport, Nick Scott, Sherrie Kelly, Caroline Bennette, Bradley Wagner, Stewart Chang, Jasmina Panovska-Griffiths, and Edward Wenger. Schematics were produced by Amanda Izzo. Supervision was provided by Daniel Klein, Michael Famulare, Greer Fowler, and Edward Wenger. The manuscript was written by Cliff Kerr, Dina Mistry, Robyn Stuart, Rafael Núñez, and Daniel Klein. Additional manuscript review and editing was performed by Katherine Rosenfeld, Jamie Cohen, and Michael Famulare.

Acknowledgements

Additional contributors to the Covasim model and this study include: from GitHub, William Fitzgerald, Hamel Husain, Cory Gwin, Julian Nadeau, Rasmus Wriedt Larsen, Aditya Sharad, and Oege de Moor; from Microsoft, William Chen, Scott Ayers, and Rolf Harms; from the Institute for Disease Modeling, Mary Fisher, Jennifer Schripsema, Dennis Chao, Christian Wiswell, Samuel Buxton, Christopher Lorton, Clinton Collins, Christopher Jones, Charles Eliot, Svetlana Titova, Dejan Lukacevic, Jeffrey Steinkraus, John Sheppard, Niket Thakkar, Roy Burstein, Robert Hart, Guillaume Chabot-Couture, Caitlin Bever, Helen Olsen, and Natalia Corona; from the Allen Institute, Natalia Orlova; from the Jet Propulsion Laboratory, Casey Handmer; from the QIMR Berghofer Medical Research Institute, Paula Sanz-Leon and James Roberts; from the Kirby Institute, Richard Gray; from the Burnet Institute, Tharindu Wickramaarachchi; from the University of California San Diego, Richard K. Belew; from the London School of Hygiene & Tropical Medicine, William Waites; and from Novosibirsk State University, Mariia Sosnovskaya. We also wish to thank the participants of the Covasim Users Group, including Julie Maher, Dean Sidelinger, and Erik Everson from the Oregon Health Authority; Samuel Mwalili and Duncan Gathungu from Jomo Kenyatta University of Agriculture and Technology; André Lin Ouédraogo from the Institute for Disease Modeling; David P. Wilson from the Bill and Melinda Gates Foundation; Edinah Mudimu, Brian Mudimu, and Chris Swanepoel from the University of South Africa; and Quang Duy Pham from the Pasteur Institute of Ho Chi Minh City.

Funding

Bill and Melinda Gates Foundation. Institutional support, including high-performance computing and library access, was provided by the Burnet Institute and the University of Sydney School of Physics.

Competing interests

Authors declare no competing interests.

Data and materials availability

The Covasim model code and documentation is fully open source and available via GitHub or via covasim.org.

References

1. Everts J. The dashboard pandemic. Dialogues in Human Geography. 2020 Jul 1;10(2):260–4.
2. Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. The Lancet Infectious Diseases. 2020 May 1;20(5):533–4.
3. Public Health Seattle King County. COVID-19 Data Dashboard [Internet]. 2020. Available from: https://www.kingcounty.gov/depts/health/covid-19/data.aspx
4. Walker PGT, Whittaker C, Watson OJ, Baguelin M, Winskill P, Hamlet A, et al. The impact of COVID-19 and strategies for mitigation and suppression in low- and middle-income countries. Science. 2020 Jun 12;eabc0035.
5. Read JM, Bridgen JR, Cummings DA, Ho A, Jewell CP. Novel coronavirus 2019-nCoV: early estimation of epidemiological parameters and epidemic predictions. medRxiv. 2020 Jan 28;2020.01.23.20018549.
6. Keeling MJ, Hollingsworth TD, Read JM. Efficacy of contact tracing for the containment of the 2019 novel coronavirus (COVID-19). J Epidemiol Community Health. 2020 Jun 22;74(10):861–6.
7. Dehning J, Zierenberg J, Spitzner FP, Wibral M, Neto JP, Wilczek M, et al. Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. Science. 2020 Jul 10;369(6500):eabb9789.
8. Giordano G, Blanchini F, Bruno R, Colaneri P, Di Filippo A, Di Matteo A, et al. Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy. Nature Medicine. 2020 Apr 22;1–6.
9. Zhao S, Chen H. Modeling the epidemic dynamics and control of COVID-19 outbreak in China. Quant Biol. 2020 Mar;1–9.
10. Ferguson N, Laydon D, Nedjati Gilani G, Imai N, Ainslie K, Baguelin M, et al. Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand [Internet]. 2020 Mar [cited 2021 Feb 7]. Available from: http://spiral.imperial.ac.uk/handle/10044/1/77482
11. Rockett RJ, Arnott A, Lam C, Sadsad R, Timms V, Gray K-A, et al. Revealing COVID-19 transmission in Australia by SARS-CoV-2 genome sequencing and agent-based modeling. Nature Medicine. 2020 Sep;26(9):1398–404.
12. Koo JR, Cook AR, Park M, Sun Y, Sun H, Lim JT, et al. Interventions to mitigate early spread of SARS-CoV-2 in Singapore: a modelling study. The Lancet Infectious Diseases. 2020 Mar 23;20(6):678–88.
13. Chao DL, Oron AP, Srikrishna D, Famulare M. Modeling layered non-pharmaceutical interventions against SARS-CoV-2 in the United States with Corvid. medRxiv. 2020 Apr 11;2020.04.08.20058487.
14. Aleta A, Martín-Corral D, Pastore y Piontti A, Ajelli M, Litvinova M, Chinazzi M, et al. Modelling the impact of testing, contact tracing and household quarantine on second waves of COVID-19. Nature Human Behaviour. 2020 Sep;4(9):964–71.
15. Kretzschmar ME, Rozhnova G, Bootsma MCJ, Boven M van, Wijgert JHHM van de, Bonten MJM. Impact of delays on effectiveness of contact tracing strategies for COVID-19: a modelling study. The Lancet Public Health. 2020 Aug 1;5(8):e452–9.
16. Kucharski AJ, Klepac P, Conlan AJK, Kissler SM, Tang ML, Fry H, et al. Effectiveness of isolation, testing, contact tracing, and physical distancing on reducing transmission of SARS-CoV-2 in

different settings: a mathematical modelling study. The Lancet Infectious Diseases. 2020 Jun 16;20(10):1151–60.
17. Blakely T, Thompson J, Carvalho N, Bablani L, Wilson N, Stevenson M. The probability of the 6-week lockdown in Victoria (commencing 9 July 2020) achieving elimination of community transmission of SARS-CoV-2. The Medical Journal of Australia. 2020 Sep 28;213(8):349-351.e1.
18. Hoertel N, Blachier M, Blanco C, Olfson M, Massetti M, Rico MS, et al. A stochastic agent-based model of the SARS-CoV-2 epidemic in France. Nature Medicine. 2020 Sep;26(9):1417–21.
19. Lau MSY, Grenfell B, Thomas M, Bryan M, Nelson K, Lopman B. Characterizing superspreading events and age-specific infectiousness of SARS-CoV-2 transmission in Georgia, USA. Proc Natl Acad Sci USA. 2020 Sep 8;117(36):22430–5.
20. Peak CM, Childs LM, Grad YH, Buckee CO. Comparing nonpharmaceutical interventions for containing emerging epidemics. Proc Natl Acad Sci U S A. 2017 Apr;114(15):4023–8.
21. Hellewell J, Abbott S, Gimma A, Bosse NI, Jarvis CI, Russell TW, et al. Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. Vol. 8, The Lancet Global Health. 2020. p. e488–96.
22. Perkins TA, Cavany SM, Moore SM, Oidtman RJ, Lerch A, Poterek M. Estimating unobserved SARS-CoV-2 infections in the United States. PNAS. 2020 Sep 8;117(36):22597–602.
23. Hinch R, Probert WJM, Nurtay A, Kendall M, Wymatt C, Hall M, et al. OpenABM-Covid19 - an agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. medRxiv. 2020 Sep 22;2020.09.16.20195925.
24. Abueg M, Hinch R, Wu N, Liu L, Probert W, Wu A, et al. Modeling the combined effect of digital exposure notification and non-pharmaceutical interventions on the COVID-19 epidemic in Washington state. medRxiv. 2020 Sep 2;2020.08.29.20184135.
25. Kerr CC, Mistry D, Stuart RM, Rosenfeld K, Hart GR, Núñez RC, et al. Controlling COVID-19 via test-trace-quarantine. medRxiv. 2020 Oct 3;2020.07.15.20154765.
26. Hamer DH, White LF, Jenkins HE, Gill CJ, Landsberg HN, Klapperich C, et al. Control of COVID-19 transmission on an urban university campus during a second wave of the pandemic. medRxiv. 2021 Feb 26;2021.02.23.21252319.
27. Cohen JA, Mistry D, Kerr CC, Klein DJ. Schools are not islands: Balancing COVID-19 risk and educational benefits using structural and temporal countermeasures. medRxiv. 2020 Sep 10;2020.09.08.20190942.
28. Pham QD, Stuart RM, Nguyen TV, Luong QC, Tran DQ, Pham TQ, et al. Lessons learned from Vietnam's COVID-19 response: the role of adaptive behavior change and testing in epidemic control. medRxiv. 2020 Dec 19;2020.12.18.20248454.
29. Panovska-Griffiths J, Kerr CC, Stuart RM, Mistry D, Klein DJ, Viner RM, et al. Determining the optimal strategy for reopening schools, the impact of test and trace interventions, and the risk of occurrence of a second COVID-19 epidemic wave in the UK: a modelling study. The Lancet Child & Adolescent Health. 2020 Nov 1;4(11):817–27.
30. Scott N, Palmer A, Delport D, Abeysuriya R, Stuart R, Kerr CC, et al. Modelling the impact of reducing control measures on the COVID-19 pandemic in a low transmission setting. The Medical Journal of Australia. 2020 Sep 2;1.
31. Shea K, Borchering RK, Probert WJM, Howerton E, Bogich TL, Li S, et al. COVID-19 reopening strategies at the county level in the face of uncertainty: Multiple Models for Outbreak Decision Support. medRxiv. 2020 Nov 5;2020.11.03.20225409.
32. Stuart RM, Abeysuriya RG, Kerr CC, Mistry D, Klein DJ, Gray R, et al. The role of masks in reducing the risk of new waves of COVID-19 in low transmission settings: a modeling study. medRxiv. 2020 Sep 3;2020.09.02.20186742.
33. Panovska-Griffiths J, Kerr CC, Waites W, Stuart RM, Mistry D, Foster D, et al. Modelling the potential impact of mask use in schools and society on COVID-19 control in the UK. medRxiv. 2020 Oct 8;2020.09.28.20202937.
34. Abeysuriya RG, Delport D, Stuart RM, Sacks-Davis R, Kerr CC, Mistry D, et al. Preventing a cluster from becoming a new wave in settings with zero community COVID-19 cases. medRxiv. 2020 Dec

22;2020.12.21.20248595.
35. Bi Q, Wu Y, Mei S, Ye C, Zou X, Zhang Z, et al. Epidemiology and transmission of COVID-19 in 391 cases and 1286 of their close contacts in Shenzhen, China: a retrospective cohort study. Vol. 20, The Lancet Infectious Diseases. 2020. p. 911–9.
36. Yang Y, Yang M, Shen C, Wang F, Yuan J, Li J, et al. Evaluating the accuracy of different respiratory specimens in the laboratory diagnosis and monitoring the viral shedding of 2019-nCoV infections. medRxiv. 2020 Feb 17;2020.02.11.20021493.
37. Lauer SA, Grantz KH, Bi Q, Jones FK, Zheng Q, Meredith HR, et al. The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application. Vol. 172, Ann Intern Med. 2020. p. 577–82.
38. Du Z, Xu X, Wu Y, Wang L, Cowling BJ, Meyers LA. Serial Interval of COVID-19 among Publicly Reported Confirmed Cases. Vol. 26, Emerg Infect Dis. 2020.
39. Nishiura H, Linton NM, Akhmetzhanov AR. Serial interval of novel coronavirus (COVID-19) infections. Vol. 93, Int J Infect Dis. 2020. p. 284–6.
40. Pung R, Chiew CJ, Young BE, Chin S, Chen MIC, Clapham HE, et al. Investigation of three clusters of COVID-19 in Singapore: implications for surveillance and response measures. Vol. 395, The Lancet. 2020. p. 1039–46.
41. Linton NM, Kobayashi T, Yang Y, Hayashi K, Akhmetzhanov AR, Jung S-M, et al. Incubation Period and Other Epidemiological Characteristics of 2019 Novel Coronavirus Infections with Right Truncation: A Statistical Analysis of Publicly Available Case Data. Vol. 9, Journal of clinical medicine. 2020. p. 538.
42. He X, Lau EHY, Wu P, Deng X, Wang J, Hao X, et al. Temporal dynamics in viral shedding and transmissibility of COVID-19. Nature Medicine. 2020 May;26(5):672–5.
43. Wang D, Hu B, Hu C, Zhu F, Liu X, Zhang J, et al. Clinical Characteristics of 138 Hospitalized Patients With 2019 Novel Coronavirus–Infected Pneumonia in Wuhan, China. JAMA. 2020 Mar 17;323(11):1061.
44. Chen J, Qi T, Liu L, Ling Y, Qian Z, Li T, et al. Clinical progression of patients with COVID-19 in Shanghai, China. Journal of Infection. 2020 May 1;80(5):e1–6.
45. Verity R, Okell LC, Dorigatti I, Winskill P, Whittaker C, Imai N, et al. Estimates of the severity of coronavirus disease 2019: a model-based analysis. Vol. 20, The Lancet Infectious Diseases. 2020. p. 669–77.
46. Wölfel R, Corman VM, Guggemos W, Seilmaier M, Zange S, Müller MA, et al. Virological assessment of hospitalized patients with COVID-2019. Vol. 581, Nature. 2020. p. 465–9.
47. Zhang J, Litvinova M, Liang Y, Wang Y, Wang W, Zhao S, et al. Changes in contact patterns shape the dynamics of the COVID-19 outbreak in China. Science. 2020 Jun 26;368(6498):1481–6.
48. O'Driscoll M, Ribeiro Dos Santos G, Wang L, Cummings DAT, Azman AS, Paireau J, et al. Age-specific mortality and immunity patterns of SARS-CoV-2. Nature. 2021 Feb;590(7844):140–5.
49. Brazeau N, Verity R, Jenks S, Fu H, Whittaker C, Winskill P, et al. Report 34: COVID-19 infection fatality ratio: estimates from seroprevalence [Internet]. 2020. Available from: https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/covid-19/report-34-ifr/
50. Ferguson NM, Laydon D, Nedjati-Gilani G, Imai N, Ainslie K, Baguelin M, et al. Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand. London: Imperial College COVID-19 Response Team, March. 2020;16.
51. Russo L, Anastassopoulou C, Tsakris A, Bifulco GN, Campana EF, Toraldo G, et al. Tracing day-zero and forecasting the COVID-19 outbreak in Lombardy, Italy: A compartmental modelling and numerical optimization approach. PLOS ONE. 2020 Oct 30;15(10):e0240649.
52. Singh R, Adhikari R. Age-structured impact of social distancing on the COVID-19 epidemic in India. arXiv:200312055 [cond-mat, q-bio] [Internet]. 2020 Mar 26 [cited 2021 Mar 28]; Available from: http://arxiv.org/abs/2003.12055
53. Lader D, Short S, Gershuny J. The time use survey, 2005. Office for National Statistics, London; 2006 Aug.
54. Cevik M, Tate M, Lloyd O, Maraolo AE, Schafers J, Ho A. SARS-CoV-2, SARS-CoV, and MERS-CoV

viral load dynamics, duration of viral shedding, and infectiousness: a systematic review and meta-analysis. The Lancet Microbe. 2021 Jan 1;2(1):e13–22.
55. He D, Zhao S, Lin Q, Zhuang Z, Cao P, Wang MH, et al. The relative transmissibility of asymptomatic COVID-19 infections among close contacts. Vol. 94, International Journal of Infectious Diseases. 2020. p. 145–7.
56. Lescure F-X, Bouadma L, Nguyen D, Parisey M, Wicky P-H, Behillil S, et al. Clinical and virological data of the first cases of COVID-19 in Europe: a case series. Lancet Infect Dis. 2020 Mar;20(6):697–706.
57. To KK-W, Tsang OT-Y, Leung W-S, Tam AR, Wu T-C, Lung DC, et al. Temporal profiles of viral load in posterior oropharyngeal saliva samples and serum antibody responses during infection by SARS-CoV-2: an observational cohort study. Lancet Infect Dis. 2020 May;20(5):565–74.
58. Zou L, Ruan F, Huang M, Liang L, Huang H, Hong Z, et al. SARS-CoV-2 Viral Load in Upper Respiratory Specimens of Infected Patients. N Engl J Med. 2020 Mar;382(12):1177–9.
59. Endo A, Centre for the Mathematical Modelling of Infectious Diseases COVID-19 Working Group, Abbott S, Kucharski AJ, Funk S. Estimating the overdispersion in COVID-19 transmission using outbreak sizes outside China. Wellcome Open Res. 2020 Apr 9;5:67.
60. Miller D, Martin MA, Harel N, Kustin T, Tirosh O, Meir M, et al. Full genome viral sequences inform patterns of SARS-CoV-2 spread into and within Israel. medRxiv. 2020 May 22;2020.05.21.20104521.
61. Ferretti L, Wymant C, Kendall M, Zhao L, Nurtay A, Abeler-Dörner L, et al. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. Science. 2020 May 8;368(6491):eabb6936.
62. Mistry D, Litvinova M, Pastore y Piontti A, Chinazzi M, Fumanelli L, Gomes MFC, et al. Inferring high-resolution human mixing patterns for disease modeling. Nature Communications. 2021 Jan 12;12(1):323.
63. Fumanelli L, Ajelli M, Manfredi P, Vespignani A, Merler S. Inferring the Structure of Social Contacts from Demographic Data in the Analysis of Infectious Diseases Spread. PLOS Computational Biology. 2012 Sep 13;8(9):e1002673.
64. Smieszek T, Barclay VC, Seeni I, Rainey JJ, Gao H, Uzicanin A, et al. How should social mixing be measured: comparing web-based survey and sensor-based methods. BMC Infect Dis. 2014 Mar;14:136.
65. United States Census Bureau [Internet]. [cited 2021 Feb 6]. Available from: https://data.census.gov/cedsci/
66. Huisman J, Smits J. Effects of Household- and District-Level Factors on Primary School Enrollment in 30 Developing Countries. World Development. 2009 Jan;37(1):179–93.
67. Prem K, Cook AR, Jit M. Projecting social contact matrices in 152 countries using contact surveys and demographic data. PLoS Comput Biol. 2017 Sep;13(9):e1005697.
68. Mossong J, Hens N, Jit M, Beutels P, Auranen K, Mikolajczyk R, et al. Social Contacts and Mixing Patterns Relevant to the Spread of Infectious Diseases. PLOS Medicine. 2008 Mar 25;5(3):e74.
69. Dodd PJ, Looker C, Plumb ID, Bond V, Schaap A, Shanaube K, et al. Age- and Sex-Specific Social Contact Patterns and Incidence of Mycobacterium tuberculosis Infection. American Journal of Epidemiology. 2016 Jan 15;183(2):156–66.
70. Ledford H, Cyranoski D, Van Noorden R. The UK has approved a COVID vaccine — here's what scientists now want to know. Nature. 2020 Dec 10;588(7837):205–6.
71. Vetter P, Kaiser L, Calmy A, Agoritsas T, Huttner A. Dexamethasone and remdesivir: finding method in the COVID-19 madness. The Lancet Microbe. 2020 Dec;1(8):e309–10.
72. Barratt H, Kirwan M. Public Health Textbook [Internet]. Health Knowledge. 2010 [cited 2020 May 9]. Available from: https://www.healthknowledge.org.uk/public-health-textbook
73. Gostic KM, McGough L, Baskerville EB, Abbott S, Joshi K, Tedijanto C, et al. Practical considerations for measuring the effective reproductive number, Rt. PLOS Computational Biology. 2020 Dec 10;16(12):e1008409.
74. Bakir ST. Compound Interest Doubling Time Rule: Extensions and Examples from Antiquities. Communications in Mathematical Finance. 2016;5(2).

75. Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, Dessimoz C. Approximate Bayesian Computation. PLOS Computational Biology. 2013 Jan 10;9(1):e1002803.
76. Mathews P. Sample size calculations: Practical methods for engineers and scientists. Mathews Malnar and Bailey; 2010.
77. Kerr CC, Dura-Bernal S, Smolinski TG, Chadderdon GL, Wilson DP. Optimization by adaptive stochastic descent. PLOS ONE. 2018;13(3).
78. Andrianakis I, Vernon IR, McCreesh N, McKinley TJ, Oakley JE, Nsubuga RN, et al. Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda. Wu H, editor. PLoS Comput Biol. 2015 Jan 8;11(1):e1003968.
79. Klinger E, Rickert D, Hasenauer J. pyABC: distributed, likelihood-free inference. Bioinformatics. 2018 Oct 15;34(20):3591–3.
80. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. In: "Optuna: A Next-generation Hyperparameter Optimization Framework", Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [Internet]. Anchorage, AK, USA: Association for Computing Machinery; 2019 [cited 2020 Jul 11]. p. 2623–31. (KDD '19). Available from: https://doi.org/10.1145/3292500.3330701
81. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020 Mar;17(3):352.
82. Oberkampf WL, Helton JC, Joslyn CA, Wojtkiewicz SF, Ferson S. Challenge problems: uncertainty in system response given uncertain parameters. Reliability Engineering & System Safety. 2004 Jul 1;85(1):11–9.
83. Willink R, Lira I. A united interpretation of different uncertainty intervals. Measurement. 2005 Jul 1;38(1):61–6.
84. Huerta-Cepas J, Serra F, Bork P. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. Mol Biol Evol. 2016 Jun;33(6):1635–8.
85. Chang S, Pierson E, Koh PW, Gerardin J, Redbird B, Grusky D, et al. Mobility network models of COVID-19 explain inequities and inform reopening. Nature. 2021 Jan;589(7840):82–7.
86. Noll NB, Aksamentov I, Druelle V, Badenhorst A, Ronzani B, Jefferies G, et al. COVID-19 Scenarios: an interactive tool to explore the spread and associated morbidity and mortality of SARS-CoV-2. medRxiv. 2020 May 12;2020.05.05.20091363.
87. Contreras S, Dehning J, Loidolt M, Zierenberg J, Spitzner FP, Urrea-Quintero JH, et al. The challenges of containing SARS-CoV-2 via test-trace-and-isolate. Nature Communications. 2021 Jan 15;12(1):378.
88. Reiner RC, Barber RM, Collins JK, Zheng P, Adolph C, Albright J, et al. Modeling COVID-19 scenarios for the United States. Nature Medicine. 2021 Jan;27(1):94–105.
89. Kucharski AJ, Andreasen V, Gog JR. Capturing the dynamics of pathogens with many strains. J Math Biol. 2016 Jan 1;72(1):1–24.
90. Kerr CC. Is epidemiology ready for Big Software? Pathogens and Disease. 2019 Feb 1;77(1):ftz006.
91. Colbourn T, Waites W, Manheim D, Foster D, Sturniolo S, Sculpher M, et al. Modelling the health and economic impacts of different testing and tracing strategies for COVID-19 in the UK. F1000Res. 2020 Dec 14;9:1454.
92. Sturniolo S, Waites W, Colbourn T, Manheim D, Panovska-Griffiths J. Testing, tracing and isolation in compartmental models. medRxiv. 2020 Dec 1;2020.05.14.20101808.
93. Holmdahl I, Buckee C. Wrong but Useful — What Covid-19 Epidemiologic Models Can and Cannot Tell Us. New England Journal of Medicine. 2020 Jul 23;383(4):303–5.
94. Ioannidis JPA, Cripps S, Tanner MA. Forecasting for COVID-19 has failed. International Journal of Forecasting. 2020 Aug;S0169207020301199.
95. Chin V, Samia NI, Marchant R, Rosen O, Ioannidis JPA, Tanner MA, et al. A case study in model failure? COVID-19 daily deaths and ICU bed utilisation predictions in New York state. Eur J Epidemiol. 2020 Aug 1;35(8):733–42.