SimpactCyan 1.0: An Open-source Simulator for Individual-Based Models in HIV Epidemiology with R and Python Interfaces

Jori Liesenborgs[1], **Diana M Hendrickx**[2], **Elise Kuylen**[3,4], **David Niyukuri**[4,5], **Niel Hens**[2,6], **and Wim Delva**[2,4,5,7,8,*]

[1]Expertise Centre for Digital Media, Hasselt University – tUL, Diepenbeek, Belgium
[2]Center for Statistics, I-BioStat, Hasselt University, Diepenbeek, Belgium
[3]MOdeling of Systems and Internet Communication (MOSAIC) research group, Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium
[4]The South African Department of Science and Technology-National Research Foundation (DST-NRF) Centre of Excellence in Epidemiological Modelling and Analysis (SACEMA), Stellenbosch University, Stellenbosch, South Africa
[5]Department of Global Health, Faculty of Medicine and Health, Stellenbosch University, Stellenbosch, South Africa
[6]Centre for Health Economics Research and Modelling Infectious Diseases and Centre for the Evaluation of Vaccination, Vaccine & Infectious Disease Institute, University of Antwerp, Antwerp, Belgium
[7]International Centre for Reproductive Health, Ghent University, Ghent, Belgium
[8]Rega Institute for Medical Research, KU Leuven, Leuven, Belgium
[*]wimdelva@gmail.com

ABSTRACT

SimpactCyan is an open-source simulator for individual-based models in HIV epidemiology. Its core algorithm is written in C++ for computational efficiency, while the R and Python interfaces aim to make the tool accessible to the fast-growing community of R and Python users. Transmission, treatment and prevention of HIV infections in dynamic sexual networks are simulated by discrete events, which include the formation and dissolution of sexual relationships, conception and birth, HIV-related and non-HIV-related death, transmission and diagnosis of HIV, and the initiation and discontinuation of HIV treatment. A generic "intervention" event allows model parameters to be changed over time, and can be used to model medical and behavioural HIV prevention programmes. Event-times for the discrete events are sampled in continuous time from user-defined hazard functions, using the modified Next Reaction Method (mNRM). First, we describe a more efficient variant of the mNRM that drives the simulator. Next, we outline key built-in features and assumptions of individual-based models formulated in SimpactCyan, and provide code snippets for how to formulate, execute and analyse models in SimpactCyan through its R and Python interfaces. Lastly, we give two examples of applications in HIV epidemiology: the first demonstrates how the software can be used to estimate the impact of progressive changes to the eligibility criteria for HIV treatment on HIV incidence. The second example illustrates the use of SimpactCyan as a data-generating tool for assessing the performance of a phylodynamic inference framework.

Keywords
Individual-based modelling, open-source, HIV epidemiology, C++, R, Python.

Introduction

In epidemiology, mathematical models are widely used to simulate progression, transmission, prevention and treatment of infectious diseases. The majority of these models are deterministic compartmental models, simulating population averages of changes in infection status and disease stages over time. However, many infectious diseases, in particular sexually transmitted diseases, are subject to high individual heterogeneity. Unlike compartmental models simulating population averages, individual-based models (IBMs) keep track of the events that happen to each individual separately, and are therefore able to take into account various sources of individual heterogeneity[1].

The ability to let population-level features of complex systems emerge from processes and events that happen to interacting individuals, is arguably the most important quality of IBMs. As the computational expense of IBMs has become less prohibitive thanks to multi-core processors and increased access to high-performance computers, there is a growing use of IBMs in infectious disease epidemiology[2]. SimpactCyan is conceived as a versatile model-building tool to address research questions in HIV epidemiology at the intersection of network and social epidemiology, computational biology, public health and policy modelling.

A large amount of general frameworks for individual-based simulations have been developed in the last decades. These platforms vary widely in terms of platform properties, usability, operating ability, pragmatics and security management, which makes it difficult to choose the most suitable framework for simulation in the context of a particular research question[3].

Current software for implementing IBMs to address questions in HIV epidemiology has several shortcomings. While some modelling tools (e.g. STDSIM for simulating transmission of HIV and other Sexually Transmitted Diseases[4]) are not open source, other IBMs (e.g. EMOD[5]) are relatively difficult to modify. Another limitation of EMOD is that it can only be used on computers running Windows 10, Windows Server 12, Windows HPC Server 12 or CentOS 7.1. Furthermore, while it has interfaces for Matlab and Python, it does not have an R interface. NetLogo models, on the other hand, are easily modifiable[3], and can be run from within the R environment[6], but are prohibitively slow for simulating large populations over the time-scale relevant for HIV epidemiology.

With a few exceptions (e.g. the MicSim Package[7]), existing simulators implement IBMs in discrete time. However, a continuous time implementation of IBMs has the advantage that it elegantly handles competing risks to multiple events. For instance, an individual may be concurrently at risk of HIV-related mortality and at risk of transmitting the virus to a partner. Evaluating the model in fixed time steps may lead to the situation where both events are scheduled to have taken place between now and the next time step. However, in reality, this is only possible if the transmission event happens first. In the continuous time model evaluation, we know exactly which of the two events is scheduled first, and logical consequences for the likelihood of subsequent events are processed along with the execution of the first event. Furthermore, events happening after short and long time periods can be included in a single simulation. In contrast, in a discrete time model, simulating events that occur on vastly different time-scales can be computationally inefficient. Frequently occurring events may require a small time step, possibly leading to the occurrence of rare events being evaluated with a much higher frequency than necessary. Another limitation of existing implementations of IBMs for dynamic sexual networks, is that they require ad-hoc decisions about who and in what order people "go out" to find partners and can "be found".

SimpactCyan is a simulator for event-driven IBMs in HIV epidemiology, evaluated in continuous time: the state of the system is updated each time an event happens. Furthermore, all possible relationships are considered simultaneously instead of sequentially.

Simpact (*SimpactWhite*) was first developed in Matlab[8–10]. Later, variants were developed as a MASON Multi-agent Simulation Toolkit in Java (*SimpactBlue*), and in Python (*SimpactyPurple*)[11]. To improve both speed and user-friendliness of the tool, we embarked on a major overhaul[12] in 2013, leading to the current version (*SimpactCyan*) that combines a computationally efficient simulation engine written in C++ with R and Python interfaces.

In this paper, we describe a more efficient variant of the modified Next Reaction Method (mNRM) that drives the simulator, we outline key built-in features and assumptions of individual-based models formulated in SimpactCyan, and provide code snippets for how to formulate, execute and analyse models in SimpactCyan through its R and Python interfaces. We end by giving two examples of applications in HIV epidemiology: the first demonstrates how the software can be used to estimate the impact that changes to the eligibility criteria for antiretroviral therapy (ART) had on HIV incidence in a hyperendemic setting. The second example illustrates the use of SimpactCyan as a data-generating tool for assessing the performance of other modelling frameworks.

Discrete events simulation algorithm

The modified Next Reaction Method (mNRM)

Event times, i.e. time points in the simulation at which events are scheduled to take place, are determined using the *modified Next Reaction Method* (mNRM)[13], a more efficient variant of the Gillespie algorithm[14–16] and the Next Reaction Method[17]. The mNRM was originally designed for simulating chemical systems with time-dependent propensities and delays, but in SimpactCyan we use it to simulate how individuals are at risk of events according to time-dependent hazard functions. In the mNRM algorithm, there is a core distinction between *internal event times* and (simulated) *real-world event times*. The internal event times determine when an event will be triggered according to the event's *internal clock*. Calling the *internal* time interval until a specific event fires ΔT, such internal time intervals are randomly sampled from an exponential distribution:

$$\mathrm{prob}(x)dx = \begin{cases} \exp(-x)dx & \text{if } x \geq 0, \\ 0 & \text{elsewhere.} \end{cases} \tag{1}$$

The event's *hazard function* $h(\bullet)$, referred to as the *propensity function* in[13], maps the internal time interval ΔT until the event fires onto Δt, a real-world time interval,

$$\Delta T = \int_{t_{\text{prev}}}^{t_{\text{prev}}+\Delta t} h(X(t'),t')dt', \tag{2}$$

where t_{prev} is the previous time an event was triggered. It is this hazard h that can depend on the state $X(t)$ of the simulation, and possibly also explicitly on time t. In SimpactCyan, the state of the simulation is made up of all the individuals in the population and their respective properties, such as their age, gender, HIV infection status, ART status, and whom they are in relationships with. This state $X(t)$ does *not* depend on time in a *continuous* manner, it only changes when an event is fired, i.e. when its internal time interval expires. Note that the formula above is for a single event, and while ΔT itself is not affected by other events, the mapping onto Δt certainly can be: other events can change the simulation state, and the hazard of the event depends on this state.

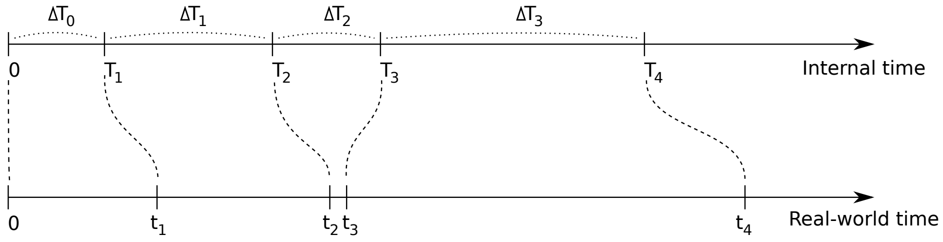


Figure 1. In the modified Next Reaction Method, intervals ΔT_i are generated independently from other events in a straightforward manner, using an exponential probability distribution (1), and are used to advance an *internal* clock T. Using the notion of a hazard function (2), these internal time intervals are mapped onto intervals Δt_i, which advance a (simulated) *real-world* time t. It is through this hazard function that interdependencies between events can be introduced.

The main idea is illustrated in Figure 1: internal time intervals are chosen from an exponential distribution, and are mapped onto real-world time intervals through hazard functions. Because hazards can depend on the simulation state and can have an explicit time dependency, this mapping can be rather complex.

While the hazard *can* cause complex behaviour, this is of course not necessarily the case. If one uses a constant hazard, this merely causes a linear scaling between internal time ΔT and real-world time Δt:

$$\Delta T = h\Delta t \quad \text{(for a constant hazard).}$$

This also illustrates that the larger the hazard, the earlier the event will fire, i.e. the real-world time interval will be smaller.

As an example, let's consider the event of forming a heterosexual relationship. At a certain time in the simulation, many formation events will be scheduled, one event for each man-woman pair that can possibly form a relationship. The internal time interval for each of these events will be drawn from the simple exponential distribution. The mapping to a real-world time at which the event will fire, is done using the hazard-based method, and this hazard depends on aspects of the simulation state as defined by the hazard function for relationship formation: how many relationships the man and woman of the candidate couple are already engaged in, what the preferred age differences with their respective partners are, etc. One can also imagine an explicit time dependency in the hazard: e.g. the hazard of forming a relationship increases as the time since the relationship became possible goes up.

While most of the events in SimpactCyan are scheduled using the exponential distribution to generate values for internal ΔT, some events are scheduled directly in real-world time. An example of this is the scheduling of the HIV 'seeding' event, i.e. the timing of introducing HIV into the population. This alternative method could still be thought of as a special case of internal and real-world time mapping. This is because if ΔT is set to the actual real-world time interval until the event fires, and the hazard is set to $h = 1$, internal and real-world time intervals match.

More efficient mNRM algorithm

Each time an event is triggered, the state of the simulation changes. Because the hazard of any event can depend on this state, in the most general version of the mNRM algorithm, one would recalculate the real-world event times of all remaining events each time an event gets triggered: this ensures that the possibly changed state is taken into account. Always recalculating all event fire times is computationally very inefficient, however. Although the state may have been changed somewhat, this change may not be relevant for many of the event hazards in use. As a result, most updated real-world event times would be the same as before.

To avoid unnecessary recalculations of event times, SimpactCyan employs a variant of the mNRM algorithm, in which each individual is linked to a list of events that involve him or her, and events that involve multiple people will appear on the lists of all of these individuals. For example, a mortality event would be present in the list of only one individual, while a relationship formation event concerns two people and would therefore appear on two such lists. Figure 2 illustrates this idea.

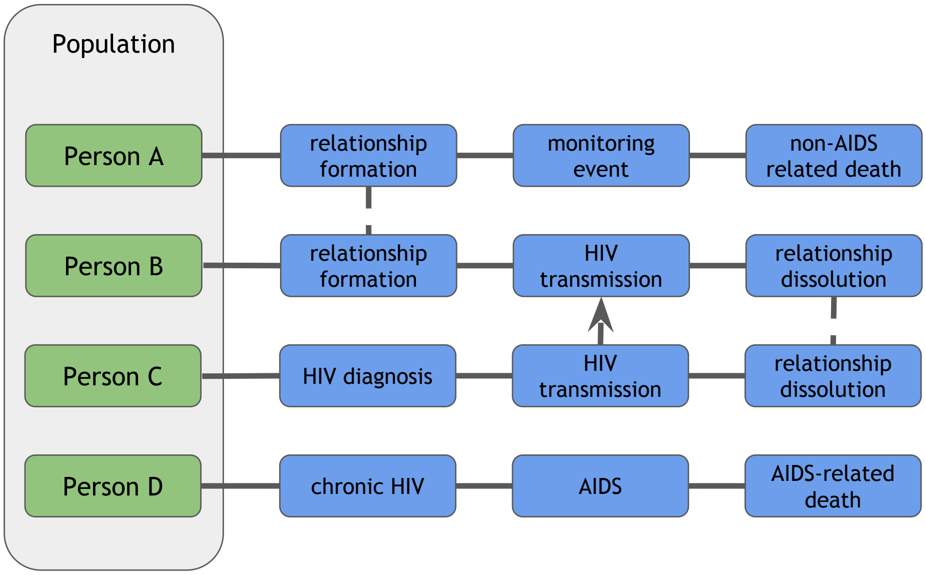


Figure 2. The Figure shows, for four different people in the population, what the next three scheduled events are. We can see that Person A and Person B will form a relationship. Only after this event triggers, relationship-related events such as HIV transmission, conception or relationship dissolution can be scheduled. Concurrent with their relationship to Person A, person B is also already in a relationship with Person C. We can see that an HIV transmission from Person C to Person B is scheduled, after Person C is diagnosed. Since person B and person C are already in a relationship, the dissolution of their relationship is also already scheduled. Person D will enter the chronic stage of HIV, after which he or she is expected to develop AIDS, and eventually die of AIDS-related complications.

When an event fires, only the properties of a very limited set of people are changed, hence one only needs to recalculate the fire times of the events in those people's lists. For example, when the event of Person A forming a relationship with Person B takes place, the real-world fire times for the events in the lists of Person A and Person B will be automatically recalculated. Apart from affecting the people in whose lists an event appears, some events can affect additional people. As an example, a birth event will only appear in the list of the pregnant woman and not in the event list of the father, because the scheduled birth should not be affected in the event of the death of the father. However, when triggered, the newborn will be listed as a child of the father. In general, the number of additionally affected people will be very small compared to the size of the population, causing only a fraction of the event fire times to be recalculated. This allows the modified algorithm to run much faster than the basic algorithm that always recalculates all event times. Furthermore, fire times of events that are present in the event lists of two individuals (e.g. relationship formation), are recalculated by only one of them.

Besides these types of events, there are also 'global' events. These events do not refer to a particular person and will modify the state in a very general way. In general, when such a global event is triggered, this causes *all* other event fire times to be recalculated. Introducing HIV into the population through an HIV seeding event is an example of a global event.

Population and events in SimpactCyan

Model populations consist of men and/or women. They can be introduced into the simulation in two ways: (i) during the initialization of the simulation, in which case individuals with certain ages (drawn from a distribution) are added to the simulation, and (ii) through the birth of new individuals during the course of the simulation run.

Once born, an individual will become sexually active when a *debut event* is triggered. If the individual is introduced into the population at the start of the simulation, and the age exceeds the debut age, this event no longer needs to be scheduled. Every person always has a 'normal' *mortality event* scheduled, which corresponds to a cause of death other than AIDS.

To get an HIV epidemic started, an *HIV seeding event* must be scheduled. When this event is triggered, a number of people in the existing population will be marked as being HIV-infected. An infected individual will go through a number of infection stages, starting with acute HIV infection. After a default duration of 3 months[18], a *chronic stage event* is triggered, moving the individual to the chronic infection stage. A fixed amount of time before dying of AIDS (15 months by default)[18], an *AIDS stage event* is triggered, marking the transition of the chronic HIV stage to the AIDS stage. Six months before the expected AIDS-related death, a *final AIDS stage event* is triggered, after which the individual is in the 'final AIDS stage'. It is assumed that one is too ill to be sexually active during this final stage[18]. When the *AIDS mortality event* is triggered, the individual dies of AIDS.

Unless the `population.msm` parameter is set to `yes` (default is `no`), relationship *formation events* will be scheduled only for heterosexual couples of men and women who are past the age of sexual debut. The `population.msm` parameter enables simulation of populations in which (some) men only form sexual relationships with other men, and/or can form relationships with with men and women. When triggered, a formation event results in the establishment of a sexual relationship, and subsequently, the female partner is at risk of falling pregnant. In that case a *conception event* will be triggered and a while later a *birth event* will take place, introducing a new individual into the population. In case one of the partners in the relationship is HIV-infected, transmission of the virus may occur. If so, a *transmission event* will fire, and the newly infected individual will go through the different infection stages as described earlier. Of course, it is also possible that the relationship will cease to exist, in which case a *dissolution event* will be triggered. Note that in the version at the time of writing, there is no mother-to-child-transmission (MTCT).

Starting ART and dropping out of treatment is managed by another sequence of events. When an individual becomes HIV-infected, either by HIV seeding or by transmission, first a *diagnosis event* is scheduled. Upon diagnosis, an *HIV monitoring event* is scheduled to monitor the progression of the HIV infection. When this event is fired, ART may be initiated, but only if the individual is both eligible (according to a CD4 cell count threshold) and willing to start HIV treatment; if not, a new monitoring event will be scheduled. If ART is initiated, no more monitoring events will be scheduled, but the individual will be at risk of discontinuing his or her HIV treatment, in which case a *dropout event* is triggered. When a person drops out of treatment, a new *diagnosis event* will be scheduled, which should be interpreted as an act of re-engagement in HIV Care[19].

Formulating, running and analysing IBMs from R or Python

Instructions for installing the core SimpactCyan program and its R interface (the Python interface is automatically installed along with the core program) can be found at `http://www.simpact.org/how-to-use-simpact/`. To set up a simulation, one needs to prepare a configuration file as a text file with key/value pairs, describing all parameters of the simulation, a snippet of which could look like this:

```
...
population.nummen     = 200
population.numwomen   = 200
population.simtime    = 40
...
```

Preparing the configuration file manually is time-consuming work however, as *all* event properties needed in a simulation must be set. To make it easier to prepare and run simulations, there is a Python module that can be used to control SimpactCyan from Python, or alternatively an R library that can be installed in R, with a similar interface. It is also possible to use a combined approach: first prepare a configuration file from within R or Python, and subsequently use this configuration to start simulations from the command-line. It can be very helpful when running simulations on a high performance computing cluster for example, where R or Python may not be available.

To use SimpactCyan from within an R session, the `RSimpactCyan` library must first be installed and loaded. This provides a `simpact.run` function that allows a simulation to be configured much more easily than using the configuration file mentioned above: instead of needing to set all parameters of a simulation, only the parameters that are different from the default values need to be specified. The full documentation of all the parameters that can be configured, what they mean and what their default values are, is found at `https://simpactcyan.readthedocs.io/en/latest/simpact_simulationdetails.html` If only the key/value pairs in the code snippet above deviate from their default values, the configuration of the simulation would simply become:

```
cfg <- list()
```

```
cfg["population.nummen"] <- 200
cfg["population.numwomen"] <- 200
cfg["population.simtime"] <- 40
```

Similarly, the Python module `pysimpactcyan` defines a `PySimpactCyan` class with a `run` member function that also needs only the settings that differ from the defaults:

```
cfg = { }
cfg["population.nummen"] = 200
cfg["population.numwomen"] = 200
cfg["population.simtime"] = 40
```

Many of the configuration values will be character strings or numbers, but for some options it is allowed to specify one of the supported one- or two-dimensional probability distributions. For example, the `birth.pregnancyduration.dist.type` is by default set to `fixed` with a value corresponding to 268/365 (simulation times are expressed in years), such that every pregnant woman would give birth after precisely 268 days. To allow for some variability (e.g. a standard deviation of 16 days), a log-normal distribution could be used instead:

```
mu.pr <- 268/365
var.pr <- (16/365)^2
cfg["birth.pregnancyduration.dist.type"] <- "lognormal"
cfg["birth.pregnancyduration.dist.lognormal.zeta"] <- log(mu.pr/sqrt(1+var.pr/mu.pr^2))
cfg["birth.pregnancyduration.dist.lognormal.sigma"] <- sqrt(log(1+var.pr/mu.pr^2))
```

Apart from using a fixed number, supported one-dimensional distributions are the beta, exponential, gamma, log-normal, normal and uniform distributions, as well as user-defined discrete distributions (e.g. based on the frequencies listed in a CSV file). For two-dimensional distributions, one can specify a fixed pair of values, or choose values from binormal or uniform distributions. Here too, user-defined discrete distributions can be specified.

Model applications

The following section discusses two example simulations that were done using SimpactCyan. The first illustrates how SimpactCyan can be used to assess the impact of progressive changes to the ART eligibility criteria in Eswatini (formerly known as Swaziland). The second illustrates the use of SimpactCyan as a data-generating tool for assessing the performance of other modelling frameworks. All code and data files necessary to reproduce the examples are available at `https://github.com/wdelva/SimpactCyanExamples`.

The impact of Early Access to ART for All on HIV incidence

In the MaxART project[20], SimpactCyan is used to estimate the likely impact of Eswatini's shift towards ”Early Access to ART for All” (EAAA) on the incidence of HIV. HIV incidence is the rate at which HIV-uninfected people acquire the infection. Such infection events are scheduled each time a relationship is formed between an HIV-infected and an HIV-uninfected individual. The hazard for the event is given by

$$\text{hazard} = \exp(a + bV^{-c} + \text{other terms}),$$

where the other terms are not enabled by default, but allow for a hazard-lowering effect of multiple ongoing relationships (so-called coital dilution[21,22], as well as a hazard-increasing effect of adolescent age among women[23]. In this formula, a, b and c are model parameters; the V value represents the current HIV viral load of the person that is already infected.

The viral load model is based upon the notion that an infected person has a specific set-point viral load, V_{sp}, which corresponds to the viral load in the chronic stage of the infection. The three parameters `person.vsp.toacute.x`, `person.vsp.toaids.x` and `person.vsp.tofinalaids.x` determine the factors by which the HIV transmission hazard should be multiplied during the initial acute stage, as well as the early and late AIDS stages. The V value in this hazard expression can therefore be different from the V_{sp} value, depending on the time since infection. The non-linear form of this hazard function was inspired by equation (9) published by Hargrove et al.[24], while the default parameter values are based on a fit to model output generated by Fraser et al.[25].

At the time of HIV acquisition, time till HIV-related death is determined, based on an early paper by Arnaout et al.[26]:

$$t_{\text{survival}} = \frac{C}{V_{\text{sp}}^{-k}} \times 10^{x},$$

In this formula, C and k are parameters that can be configured by the user if desired; the x parameter (which defaults to zero) is person-specific, and its distribution can be configured to control the amount of variation in post-HIV infection survival times among people with the same set-point viral load.

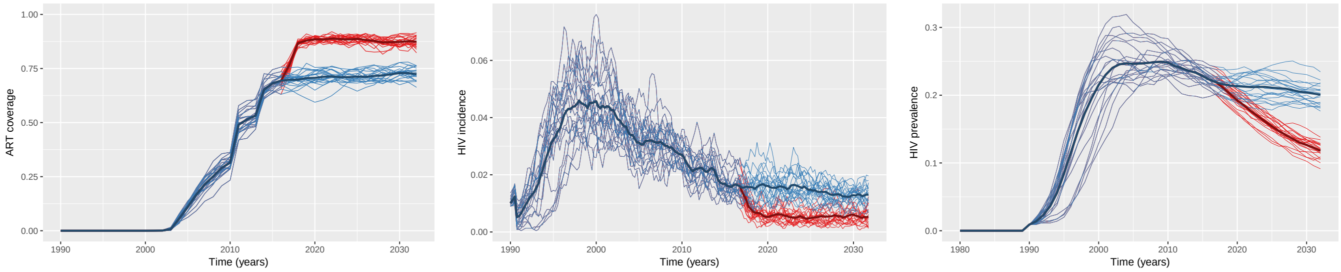
The set-point viral load value allocated to a newly infected individual is partially determined by that of their infector, i.e. some heritability of set-point viral load is assumed[27]. This is done by using a two-dimensional distribution

$$\mathrm{prob}(V_{\mathrm{sp.infector}}, V_{\mathrm{sp.infected}}),$$

of which the parameters can be chosen using the configuration values. This is subsequently used to obtain the conditional probability when fixing the initial V_{sp} value for a person that becomes infected due to the transmission event. To choose the initial set-point viral loads for 'seed infections', a marginal probability distribution is used, however.

ART initiation affects both the expected time till HIV-related death and the infectiousness of the person on ART. As soon as ART is started, the log10 viral load is assumed to drop by a user-defined fraction, and the updated current viral load is used to re-calculate t_{survival}. In the simulations of which key output is shown in Figure 3, we assume that upon ART initiation, the log10 viral load drops by 70%, effectively rendering the viral load "undetectable" for most ART clients. Via so-called intervention events, most model parameters can be changed at arbitrary points in time during the simulation. However, person-specific parameter values (e.g. the probability of accepting ART if ART-eligible) and some event-times (e.g. time of non-HIV-related death) are determined at the time the individual is introduced into the population (at the start of the simulation or at birth). Hence, changing related parameters through an intervention event would only affect individuals born into the population after this intervention event, and not the extant population.

In this example, intervention events allowed us to assume that ART was gradually introduced around the year 2000, and that the CD4 cell count threshold for ART eligibility progressively shifted towards ever more inclusive criteria, alongside a decreasing lagtime between HIV infection and HIV diagnosis. These assumptions hold in both the "Status Quo" scenario and the "Early Access to ART for All" (EAAA) scenario. In the EAAA scenario, however, an additional policy change is modelled: a policy of immediate access to ART for all people infected with HIV is adopted from October 2016. In the alternative scenario, the CD4 cell count threshold for ART eligibility stays at 500 cells/microliter from mid 2013 onwards. Extra intervention events are added in both scenarios to model a moderate reduction in sexual risk behaviour (rate of partner turnover) that took place in the period 2000 - 2003.



(a) ART coverage

(b) HIV incidence

(c) HIV prevalence

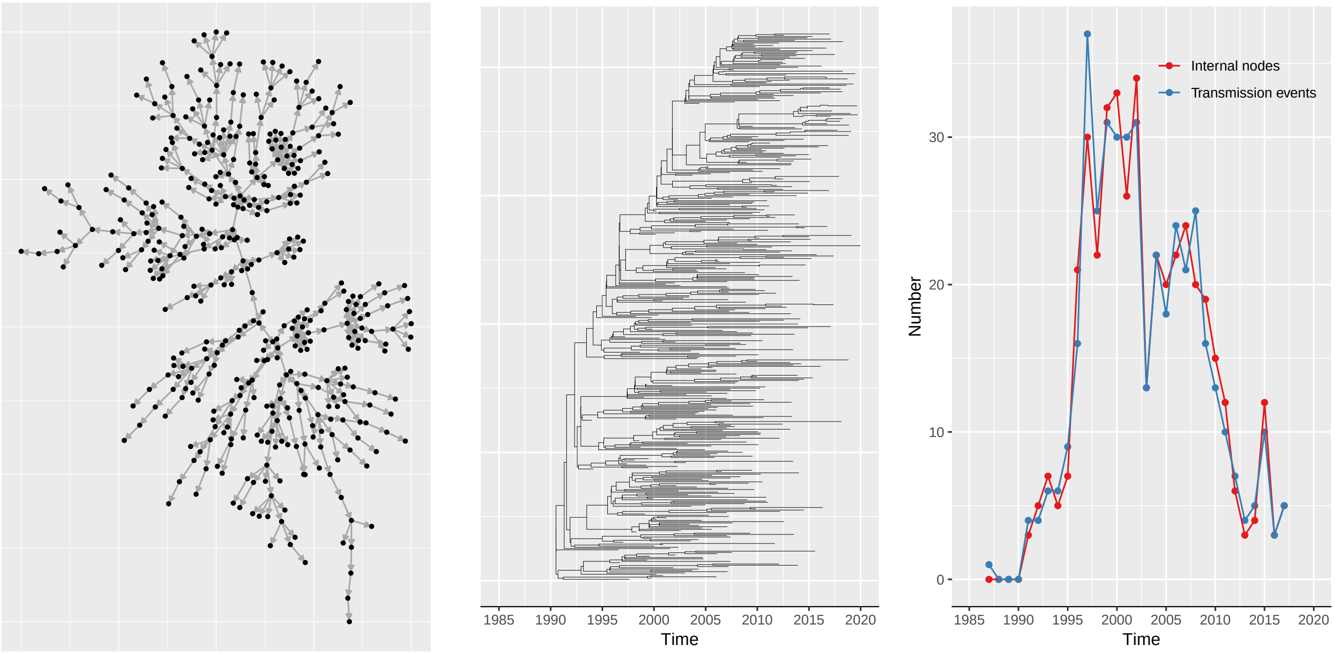
Figure 3. Programmatic and epidemic projections under a "Status Quo" (blue) and "Early Access to ART for All" (red) scenario for the roll-out of a nation-wide ART programme.

SimpactCyan as a data-generating benchmarking tool

The second use case illustrates how SimpactCyan can be used as a data-generating tool for benchmarking the performance of other modelling frameworks. Phylogenetic models have been used to infer properties of epidemics from reconstructed phylogenetic trees, including time-trends in HIV incidence rates[28,29] and the age-mixing pattern in HIV transmission clusters[30]. Yet, as the truth is typically unknown, it is difficult to assess the validity of these novel modelling frameworks, or document their sensitivity to breaches in the models' assumptions. For instance, phylogenetic inference methods typically assume that HIV sequence data are available for the majority of HIV-positive people, and that the individuals for whom the viral genome was sequenced, are a random subset of all HIV infected people. However, in many settings, neither of these assumptions is met. In Figure 4 we illustrate the basic idea of SimpactCyan as a data-generating tool for benchmarking.

First, we simulate an emerging HIV epidemic. Panel (a) shows the cumulative HIV transmission network of the epidemic, linking all individuals who got infected with HIV by the end of the simulation. Next, assuming that HIV transmission events correspond to branching points in the phylogeny, we convert the transmission network into its corresponding phylogenetic tree.

Using the Seq-Gen program[31], we simulate the viral evolution along this phylogeny, assuming a generalised time-reversible substitution model[32] with a gamma-invariable mixture model for rate heterogeneity among sites. In this way, we generate synthetic HIV sequence data. Lastly, we feed these sequences into the FastTree 2 program[33] and the treedater R package[34] to reconstruct the time-resolved phylogenetic tree, shown in panel (b). If all people ever infected are included in the sequence dataset and the same molecular evolution model is used to generate the sequence data and to reconstruct the phylogenetic tree, the timing of the internal nodes in the reconstructed tree should correspond with the timing of the simulated HIV transmission events, as shown in panel (c). Now that we have established validity of this phylogenetic inference approach under ideal circumstances, we can examine the performance of the inference method under alternative scenarios in which some of the viral sequence data are missing completely at random (MCAR), missing at random (MAR) or not at random (MNAR). Through simulation-based sensitivity analyses, we could quantify how the accuracy of epidemiological characteristics inferred by the phylodynamic method depends on the magnitude of the missing data problem and the strength of the correlations between the probability of sequences being missing and covariates such as age, indicators of sexual risk behaviour or calendar time.



(a) HIV transmission network

(b) Time-resolved phylogenetic tree

(c) Distribution of internal nodes in the reconstructed phylogenetic tree (red) and simulated HIV transmission events (blue)

Figure 4. The molecular evolution of HIV viral strains is simulated across an HIV transmission network using Seq-Gen[31] (a). Next, the synthetic HIV sequence data are used to reconstruct the time-resolved phylogenetic tree with FastTree 2[33] and the treedater R package[34] (b). Under ideal circumstances of complete sampling of the transmission network and correct specification of the model for viral evolution, the timing of the internal nodes in the reconstructed phylogenetic tree (red) corresponds nearly perfectly with the timing of the simulated HIV transmission events (blue) (c).

Future directions

Ongoing developments of SimpactCyan include the addition of events for the transmission and treatment of other sexually transmitted infections such as Herpes Simplex Virus 2 (HSV-2) and Hepatitis C Virus (HCV), as well as additional events for parenteral and mother-to-child transmission of HIV and co-infections, to allow studies of HIV transmission in injecting drug users (IDU) and children. We also plan to extend the software by enabling explicit modelling of correlation between sexual risk behaviour and health seeking behaviour. This is in response to recent evidence to suggest that high sexual risk behaviour is associated with a lower likelihood to be aware of one's HIV infection, and a lower likelihood of being virally suppressed among people who know they are HIV positive[35].

Conceived as a flexible open-source, open access tool, rather than a proprietary asset, SimpactCyan's extensions and applications should not solely come from its original developers. Instead, we want to position this simulator as a vehicle for open science in HIV epidemiology. Therefore, others are encouraged to use it for the development of their own IBMs, as the starting point for their own simulation engine, as a data-generating and/or benchmarking tool in methodological research, or for educational purposes.

Acknowledgements

The authors thank Jonathan Dushoff and Roxanne Beauclair for valuable comments on earlier drafts of this manuscript.

Funding sources

This work was supported by grants G091210N, G0B4314N and W002514N from the Research Foundation – Flanders (FWO), grant ZEIN2010PR375 from the Flemish Interuniversity Council (VLIR), grant 100014 from NRF-TWAS, and a grant from the Dutch Postcode Lottery.

References

1. Railsback, S. F. & Grimm, V. *Agent-based and individual-based modeling: a practical introduction* (Princeton university press, 2011).

2. Willem, L., Verelst, F., Bilcke, J., Hens, N. & Beutels, P. Lessons from a decade of individual-based models for infectious disease transmission: a systematic review (2006-2015). *BMC Infect. Dis.* **17**, 612 (2017). DOI 10.1186/s12879-017-2699-8.

3. Kravari, K. & Bassiliades, N. A survey of agent platforms. *J. Artif. Soc. Soc. Simul.* **18**, 11 (2015).

4. Bakker, R. *et al.* Stdsim: A microsimulation model for decision support in the control of hiv and other stds. *Sex. Transm. Dis.* **27**, 652 (2000).

5. Bershteyn, A. *et al.* Implementation and applications of emod, an individual-based multi-disease modeling platform. *Pathog. Dis.* **76** (2018). DOI 10.1093/femspd/fty059.

6. Thiele, J. C. R marries netlogo: Introduction to the rnetlogo package. *J. Stat. Softw.* **58** (2014). DOI 10.18637/jss.v058.i02.

7. Zinn, S. The micsim package of r: An entry-level toolkit for continuous-time microsimulation. *Int. J. Microsimulation* **7**, 3–32 (2014).

8. Tolentino, S. L., Meng, F. & Delva, W. A simulation-based method for efficient resource allocation of combination hiv prevention. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (Cannes, France, 5-7 March 2013)*, 31–40 (ICST, Brussels, Belgium, 2013).

9. Meng, F., Hummeling, R., Tolentino, S. L., Hens, N. & Delva, W. Modelling the impact of alternative treatment strategies on hiv prevalence in south africa: a simulation study. In *6th South African AIDS Conference* (Durban, South Africa, 2013).

10. Delva, W. *et al.* Age mixing and sustained hiv epidemics: not the size but the variation of age gaps counts. In *Epidemics4 – 4th International Conference on Infectious Disease Dynamics* (Amsterdam, The Netherlands, 2013).

11. Tolentino, S. L. *Effective and efficient algorithms for simulating sexually transmitted diseases*. Ph.D. thesis, University of Iowa (2014).

12. Meng, F., Liesenborgs, J., Delva, W., Van Reeth, F. & Hens, N. Simpact cyan: accelerating agent-based, continuous time simulation of hiv transmission on vsc. In *VSC Users Day 2014* (Brussels, Belgium, 2014).

13. Anderson, D. F. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The J. Chem. Phys.* **127** (2007). DOI http://dx.doi.org/10.1063/1.2799998.

14. Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* **22**, 403 – 434 (1976). DOI http://dx.doi.org/10.1016/0021-9991(76)90041-3.

15. Bartlett, M. S. Stochastic Processes or the Statistics of Change. *J. Royal Stat. Soc. Ser. C (Applied Stat.* **2**, 44–64 (1953). DOI 10.2307/2985327.

16. Doob, J. L. Topics in the theory of markoff chains. *Transactions Am. Math. Soc.* **52**, 37–64 (1942). DOI 10.1090/S0002-9947-1942-0006633-7.

17. Gibson, M. A. & Bruck, J. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The J. Phys. Chem. A* **104**, 1876–1889 (2000). DOI 10.1021/jp993732q. http://dx.doi.org/10.1021/jp993732q.

18. Hollingsworth, T. D., Anderson, R. M. & Fraser, C. HIV-1 Transmission, by Stage of Infection. *The J. Infect. Dis.* **198**, 687–693 (2008). DOI 10.1086/590501.

19. Grimes, R. M., Hallmark, C. J., Watkins, K. L., Agarwal, S. & McNeese, M. L. Re-engagement in hiv care: A clinical and public health priority. *J AIDS Clin Res* **7**, 1–7 (2016). DOI 10.4172/2155-6113.1000543.

20. Walsh, F. J. *et al.* Impact of early initiation versus national standard of care of antiretroviral therapy in Swaziland ' s public sector health system : study protocol for a stepped-wedge randomized trial. *Trials* **18**, 1–10 (2017). DOI 10.1186/s13063-017-2128-8.

21. Delva, W. *et al.* Coital frequency and condom use in monogamous and concurrent sexual relationships in cape town, south africa. *J. Int. AIDS Soc.* **16** (2013).

22. Sawers, L., Isaac, A. G. & Stillwaggon, E. Hiv and concurrent sexual partnerships: modelling the role of coital dilution. *J. Int. AIDS Soc.* **14**, 44 (2011).

23. Yi, T. J., Shannon, B., Prodger, J., McKinnon, L. & Kaul, R. Genital immunology and HIV susceptibility in young women. *Am. J. Reproductive Immunol.* **69**, 74–79 (2013). DOI 10.1111/aji.12035.

24. Hargrove, J., Eastwood, H., Mahiane, G. & van Schalkwyk, C. How should we best estimate the mean recency duration for the bed method? *PLOS ONE* **7**, 1–12 (2012). DOI 10.1371/journal.pone.0049661.

25. Fraser, C., Hollingsworth, T. D., Chapman, R., de Wolf, F. & Hanage, W. P. Variation in hiv-1 set-point viral load: Epidemiological analysis and an evolutionary hypothesis. *Proc. Natl. Acad. Sci.* **104**, 17441–17446 (2007). DOI 10.1073/pnas.0708559104. http://www.pnas.org/content/104/44/17441.full.pdf.

26. Arnaout, R. A. *et al.* A simple relationship between viral load and survival time in hiv-1 infection. *Proc. Natl. Acad. Sci.* **96**, 11549–11553 (1999). DOI 10.1073/pnas.96.20.11549. http://www.pnas.org/content/96/20/11549.full.pdf.

27. Fraser, C. *et al.* Virulence and pathogenesis of hiv-1 infection: an evolutionary perspective. *Science* **343**, 1243727 (2014).

28. Rasmussen, D. A., Volz, E. M. & Koelle, K. Phylodynamic Inference for Structured Epidemiological Models. *PLoS Comput. Biol.* **10** (2014). DOI 10.1371/journal.pcbi.1003570.

29. Lewis, F., Hughes, G. J., Rambaut, A., Pozniak, A. & Leigh Brown, A. J. Episodic sexual transmission of HIV revealed by molecular phylodynamics. *PLoS Medicine* **5**, 0392–0402 (2008). DOI 10.1371/journal.pmed.0050050.

30. de Oliveira, T. *et al.* Transmission networks and risk of HIV infection in KwaZulu-Natal, South Africa: a community-wide phylogenetic study. *The Lancet HIV* **3018**, 1–10 (2016). DOI 10.1016/S2352-3018(16)30186-2.

31. Rambaut, A. & Grassly, N. C. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. applications biosciences : CABIOS* **13**, 235–8 (1997). DOI 10.1093/BIOINFORMATICS/13.3.235.

32. Tavaré, S. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lect. on mathematics life sciences. Vol. 17* 57–86 (1986). DOI citeulike-article-id:4801403.

33. Price, M. N., Dehal, P. S. & Arkin, A. P. FastTree 2 - Approximately maximum-likelihood trees for large alignments. *PLoS ONE* **5** (2010). DOI 10.1371/journal.pone.0009490. Price,MorganN.,2010,FastTree2.

34. Volz, E. M. & Frost, S. D. W. Scalable relaxed clock phylogenetic dating. *Virus Evol.* **3** (2017). DOI 10.1093/ve/vex025.

35. Huerga, H. *et al.* Higher risk sexual behaviour is associated with unawareness of hiv-positivity and lack of viral suppression –implications for treatment as prevention. *Sci. Reports* **7**, 16117 (2017). DOI 10.1038/s41598-017-16382-6.