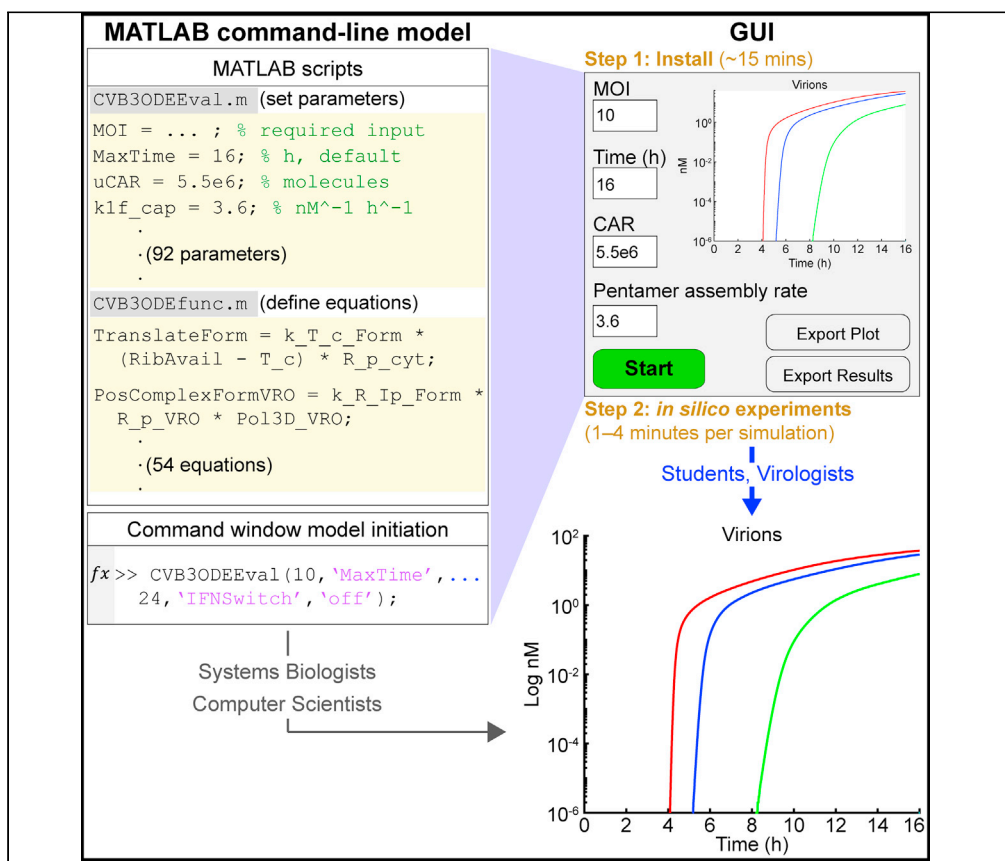


Protocol

Simulating coxsackievirus B3 infection with an accessible computational model of its complete kinetics



Cameron D. Griffiths, Andrew J. Sweatt, Kevin A. Janes

cg3gh@virginia.edu (C.D.G.)
kjanes@virginia.edu (K.A.J.)

Highlights

Tool for simulating coxsackievirus B3 infections with a graphical user interface

Extensive functionality for parameter changes, data display, and export

Installations available for Windows, MacOS, or Linux

We describe how to use a publicly available computational model for coxsackievirus B3 (CVB3) infection that we recast as a graphical user interface (GUI). The GUI-based implementation enables non-computationalists to incorporate systems-biology modeling into their research and teaching. The model simulates the full life cycle of CVB3, including the host antiviral response, and includes 44 alterable parameters. The model simplifies some viral life cycle processes to improve interpretability and utility when performing *in silico* experiments.

Griffiths et al., STAR Protocols
2, 100940
December 17, 2021 © 2021
The Author(s).
<https://doi.org/10.1016/j.xpro.2021.100940>



Protocol

Simulating coxsackievirus B3 infection with an accessible computational model of its complete kinetics

Cameron D. Griffiths,^{1,3,4,*} Andrew J. Sweatt,^{1,3} and Kevin A. Janes^{1,2,5,*}¹Department of Biomedical Engineering, University of Virginia, Charlottesville, VA 22908, USA²Department of Biochemistry & Molecular Genetics, University of Virginia, Charlottesville, VA 22908, USA³These authors contributed equally⁴Technical contact⁵Lead contact*Correspondence: cg3gh@virginia.edu (C.D.G.), kjanes@virginia.edu (K.A.J.)
<https://doi.org/10.1016/j.xpro.2021.100940>

SUMMARY

We describe how to use a publicly available computational model for coxsackievirus B3 (CVB3) infection that we recast as a graphical user interface (GUI). The GUI-based implementation enables non-computationalists to incorporate systems-biology modeling into their research and teaching. The model simulates the full life cycle of CVB3, including the host antiviral response, and includes 44 alterable parameters. The model simplifies some viral life cycle processes to improve interpretability and utility when performing *in silico* experiments. For complete details on the use and execution of this protocol, please refer to Lopacinski et al. (2021).

BEFORE YOU BEGIN

Background and implementation

Computational models are essential tools for hypothesis generation and testing in systems biology (Bruggeman and Westerhoff, 2007). However, most computational models have limited documentation and require knowledge of structured programming to operate. To overcome this barrier for non-computationalists, we developed a graphical user interface (GUI) version of a publicly available model of coxsackievirus B3 (CVB3) infection (Lopacinski et al., 2021). CVB3 is a member of the Picornavirus family of non-enveloped, positive-strand RNA viruses. Picornaviruses are a well-studied group of viruses, with fully delineated life cycles from viral docking and entry to cell lysis (Racaniello, 2013). After entry into the cytoplasm, the compact picornaviral genome is rapidly translated as a single polypeptide, yielding an equal proportion of viral proteins. From a computational modeling perspective, the abundance of quantitative information and one-to-one protein stoichiometry make it plausible to define a complete kinetic model of CVB3 infection. Complete kinetic models maintain stoichiometric constraints and seek to explain how reactants are converted to products from start to finish (Lopacinski et al., 2021). The GUI maintains the predictive and explanatory power of the CVB3 complete kinetic model and adds user-friendliness for non-computationalists.

The model encodes the entire CVB3 lifecycle as a system of 54 differential equations (Lopacinski et al., 2021). To understand many of the input parameters and the model output, it is important to be familiar with the CVB3 life cycle. The model breaks down the viral life cycle into three main modules: delivery, replication, and encapsidation. The modules are overlaid with a set of negative feedbacks that capture the effects of several interferon-stimulated genes (ISGs) that approximate the innate immune response of the host cell.



Delivery module

CVB3 uses a multi-step process to enter a host cell. The virus first binds to decay-accelerating factor (DAF), a protein receptor on the cell surface (Shafren et al., 1995). Then, CVB3 translocates to tight junctions and binds to a second receptor called coxsackievirus and adenovirus receptor (CAR) (Bergelson et al., 1997). After binding CAR, CVB3 is internalized via endocytosis, and the positive-sense RNA genome of the virus is delivered into the cytoplasm (Coyné and Bergelson, 2006). In addition to live particles, stock preparations of CVB3 also contain many defective RNA-filled particles that are unable to replicate. For CVB3, the live-to-defective particle ratio is approximately 1:800, depending on the viral stock (Lopacinski et al., 2021). In the model, defective particles follow a delivery process similar to live particles, but the defective RNA is not translated or replicated.

Replication module

Once in the cytoplasm, the CVB3 genome is translated by host ribosomes into a single polyprotein in a cap-independent manner (Racaniello, 2013). The CVB3 genome is simultaneously bound by multiple ribosomes (polysome), forming a translation complex (Kempf and Barton, 2008). After translation, the polyprotein matures through a series of cleavage events into individual viral proteins, including the viral RNA-dependent RNA-polymerase ($3D^{pol}$), two proteases ($2A^{pro}$ and $3C^{pro}$), an ATPase ($2C^{ATPase}$), and structural capsid proteins (VP1–4) (Racaniello, 2013). In parallel, the positive-sense CVB3 genome is replicated into full-length negative-sense genomes by the viral $3D^{pol}$. The negative-sense genomes are subsequently replicated by $3D^{pol}$ to generate more positive-sense genomes (Racaniello, 2013). During replication, CVB3 hydrophobic proteins (including $2C^{ATPase}$) recruit intracellular membranes to form viral replication organelles (VROs) (Melia et al., 2019; Miller and Krijnse-Locker, 2008). The VROs serve as a platform for replication, concentrating viral proteins and accelerating replication.

Encapsidation module

To create new viral particles, CVB3 capsid proteins self-assemble with positive-sense viral RNA. First, three capsid proteins (VP0 [the precursor for VP2 and VP4], VP1, and VP3) bind to form a capsid protomer (Jiang et al., 2014). In the model, a protomer contributes immediately to one fifth of a self-assembled pentamer. Twelve pentamers are assumed to bind sequentially to form a mature viral particle. During pentamer recruitment, a positive-sense viral RNA strand may or may not be bound to a pentamer. If viral RNA is bound when the twelfth pentamer is added in the model, a filled (infectious) virion is formed. Otherwise, an empty (non-infectious) virion is formed. At any point during capsid formation, pentamers or viral RNA may unbind, but once fully assembled, the model assumes that mature virions do not disassemble into individual pentamers.

Feedbacks involving the innate immune response

Host cells have mechanisms to detect infecting viruses and mount an antiviral response through a series of events that shut down transcription and translation. Reciprocally, viruses encode countermeasures to block the antiviral response. For CVB3, melanoma differentiation-associated protein 5 (MDA5) recognizes double-stranded RNA formed during CVB3 replication (Andrejeva et al., 2004; Kang et al., 2002). MDA5 bound to double-stranded RNA signals through mitochondrial antiviral-signaling protein (MAVS) to stimulate type I interferon (IFN) production (Kawai et al., 2005; Seth et al., 2005). Type I IFN is then secreted, stimulating the infected cell (as well as nearby cells) to induce ISG expression (Sadler and Williams, 2008). CVB3 proteases $2A^{pro}$ and $3C^{pro}$ antagonize the innate immune response by cleaving host-cell proteins, including MAVS (Feng et al., 2014; Mukherjee et al., 2011). Furthermore, $2A^{pro}$ cleaves eukaryotic translation initiation factor 4G (eIF4G) (Etchison et al., 1982), which prevents translation of host mRNAs but does not affect translation of the CVB3 genome.

The model encodes the innate immune response as negative feedbacks that inhibit translation, inactivate proteases, and accelerates RNA degradation. Viral antagonism is encoded as a counteracting negative feedback that decreases the host-cell ability to initiate an immune response.

Modeling implementation

The published model is publicly available as a MATLAB package at <https://github.com/JanesLab/CompleteKinetics-CVB3>. To address the drawback that the published model requires structured programming knowledge and commercial software, we built a standalone, open-source version of the model that runs simulations behind a GUI. The protocol below describes the installation and use of the CVB3 complete kinetics GUI; more-advanced users are referred to the original research publication describing the MATLAB package (Lopacinski et al., 2021).

Install the graphical user interface (GUI)

⌚ Timing: 5–30 min

The CVB3 complete kinetics GUI requires installation of MATLAB Runtime version 9.10 (R2021a; 724 MB download size). The CVB3 GUI and MATLAB Runtime software can be installed automatically or manually on Windows or MacOS operating systems. The automatic installation uses an installation wizard to install both MATLAB Runtime (if needed) and the GUI. The manual installation is provided as an alternative, where MATLAB Runtime is separately installed and the GUI is run directly from the .exe (Windows) or .app (MacOS) file. The files for the following download steps can be found on the JanesLab Github site (see [key resources table](#)).

Note: For Linux users, we compiled the GUI on a Centos distribution and made the compiled application available on the JanesLab Github site (see [key resources table](#)). Installing and running the GUI on Linux requires knowledge of command line-based scripting, which is beyond the scope of this protocol. The instructions for installing and running the GUI can be found in the readme file provided in the Linux folder on the JanesLab Github site.

1. Windows Automatic Installation Instructions
 - a. Download CVB3_AppInstaller_web.exe, and then double click to run the program.
 - b. Click "Yes" to allow this app from an unknown publisher to make changes to your device.
 - c. Click "Next >" to advance past the installation start page.
 - d. If desired, check "Add a shortcut to the desktop".
 - e. Click "Next >" to confirm the CVB3 GUI installation folder selection.
 - f. At this point, the installer checks if MATLAB Runtime is installed. If not, perform the following steps:
 - i. Click "Next >" to confirm the MATLAB Runtime installation folder selection.
 - ii. Check "Yes" then click "Next >" to agree to the MATLAB Runtime user agreement.
 - g. Click "Install >" to install MATLAB Runtime (if needed) and the CVB3 GUI.
 - h. Click "Finish" to close the installer.
 - i. Run the CVB3 GUI like any other program.
2. Windows Manual Installation Instructions
 - a. Verify that version 9.10 (R2021a) of MATLAB Runtime is installed. If installed, MATLAB Runtime can be found in the "Add or remove programs" list, which can be found by searching for "Add or remove programs" in the computer's search bar.
 - i. If MATLAB Runtime is not installed, we advise following the "Windows Automatic Installation Instructions" to download MATLAB Runtime and the GUI simultaneously. Otherwise, download and install the appropriate version of MATLAB Runtime for R2021a from the following link on the MathWorks website: <https://www.mathworks.com/products/compiler/mcr/index.html>.
 - b. Download the CVB3_GUI_Manual_Install folder to a desired location on the hard drive.
 - c. Run the CVB3 GUI by double-clicking the CVB3_GUI.exe icon.
3. MacOS Automatic Installation Instructions
 - a. Download CVB3_AppInstaller_web.app.zip.

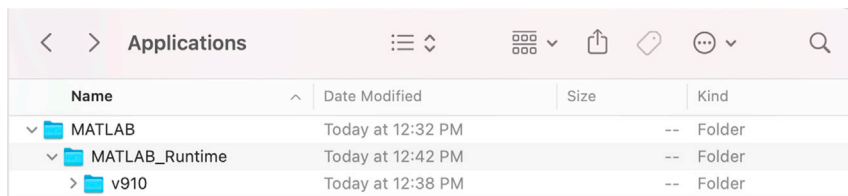


Figure 1. MacOS folder setup for MATLAB runtime

For the CVB3 model to run correctly on MacOS, MATLAB Runtime must be installed in the MATLAB folder within the Applications directory. The v910 folder within “MATLAB Runtime” is set up automatically when installing MATLAB Runtime.

- b. Right click on the file and select “Open” to open the installer.
 - i. The following message may appear: “macOS cannot verify the developer of “CVB3_ApplInstaller_web” Are you sure you want to open it?”. Click “Open” to proceed with the installation.
 - ii. If needed, enter the administrator User Name and Password and click “OK”.
- c. Click “Next >” to advance past the installation start page.
- d. Choose an installation folder and click “Next >” to confirm the CVB3 GUI installation folder selection.
- e. At this point, the installer checks if MATLAB Runtime is installed. If not, perform the following steps:
 - i. Choose an installation folder for MATLAB Runtime and click “Next >” to confirm the MATLAB Runtime installation folder selection.

Note: MATLAB Runtime must be installed in a folder named “MATLAB” in the Applications directory (Figure 1). If the application is installed in another directory, copy it to the Applications directory.

- ii. Check “Yes” then click “Next >” to agree to the MATLAB Runtime user agreement.
 - f. Click “Install >” to install MATLAB Runtime (if needed) and the CVB3 GUI.
 - g. Click “Finish” to close the installer.
 - h. Open the Janes_Lab folder (in the install location from step d—usually “Applications”).
 - i. Open the CVB3_GUI folder.
 - j. Open the application folder.
 - k. If desired, make an alias of CVB3_GUI.app for easier access.
 - l. Run the CVB3 GUI by double-clicking the CVB3_GUI.app icon.
4. MacOS Manual Installation Instructions
- a. Verify that version 9.10 (R2021a) of MATLAB Runtime is installed. If installed, MATLAB Runtime can be found in the Applications directory.
 - i. If MATLAB Runtime is not installed, we advise following the “MacOS Automatic Installation Instructions” to download MATLAB Runtime and the GUI simultaneously. Otherwise, download and install the appropriate version of MATLAB Runtime for R2021a from the following link on the MathWorks website: <https://www.mathworks.com/products/compiler/mcr/index.html>.

Note: MATLAB Runtime must be installed in a folder named “MATLAB” in the Applications directory (Figure 1). If the application is installed in another directory, copy it to the Applications directory.

- b. Download the CVB3_GUI_Manual_Install folder to a desired location on the hard drive.
- c. Open the CVB3_GUI_Manual_Install folder.
- d. Double click on the CVB3_GUI.app.zip icon.

- e. Run the CVB3 GUI by right clicking the CVB3_GUI icon and selecting “Open”.
 - i. The following message may appear: “macOS cannot verify the developer of “CVB3_AppInstaller_web”. Are you sure you want to open it?”. Click “Open” to proceed with running the CVB3 GUI.
 - ii. If needed, enter the administrator User Name and Password and click “OK”.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
CVB3 complete kinetics computational model	Lopacinski et al. (2021)	https://github.com/JanesLab/CompleteKinetics-CVB3
CVB3 complete kinetics GUI	This paper	https://github.com/JanesLab/CompleteKinetics-CVB3-GUI
MATLAB R2021a	MathWorks	https://www.mathworks.com/products/matlab.html
MATLAB Runtime version 9.10 (R2021a)	MathWorks	https://www.mathworks.com/products/compiler/matlab-runtime.html

STEP-BY-STEP METHOD DETAILS

Basic operations

⌚ Timing: 1–2 min per simulation

(Simulation time may vary with computer processing speed and non-default parameter sets.)

This protocol describes the basic model functions to provide a starting point for further simulations. The program runs simulations behind the GUI and stores the concentration of 59 output species as a function of time (e.g., virions, total single-stranded RNA, positive-sense RNA, viral protein, and interferon stimulated proteins). The user selects one species to be plotted on the main GUI figure. When multiple simulations are run sequentially, the data from previous simulations are saved until one of three things occurs: 1) the Clear Data/Plot button (General tab) is pressed, 2) the simulation mode is switched (Options tab), or 3) a sensitivity analysis is enabled or disabled (Options tab). All simulations that have not been cleared are referred to as active simulations.

To demonstrate basic operations, this protocol begins with a default infection for 16 h at a multiplicity of infection (MOI) of 10, with all parameters set to literature or experimental estimates (Lopacinski et al., 2021). At the end of this outline, the reader should be able to simulate infections, change key parameters, and export results. Descriptions of all buttons on the main GUI page are found in Table 1.

1. To begin the default simulation, click Start. The median virion concentration over time is automatically plotted along with the upper quantile (default: 95%) and lower quantile (default: 5%) (Figure 2A).

Note: The upper and lower quantiles are plotted because the CVB3 complete kinetics GUI performs multiple runs—determined by Number of runs (Options tab)—with slightly different parameter sets for a single simulation. See the [quantification and statistical analysis](#) section for details on the variation between runs.

2. To infect with half as much virus, change the number in the MOI box to 5 and click Start. The new simulation is overlaid on the previous simulation in a new color (Figure 2B, blue).

Table 1. General tab input options

Input name	Default value	Notes
MOI	10	Multiplicity of infection.
Time (h)	16	Timepoint (in hours) to end the simulation.
Antiviral Response	On	Enables/disables the interferon (IFN) response.
Viral Antagonism	On	Enables/disables viral antagonism of the host IFN response.
IFN Stimulation	Off	Enables/disables exogenous IFN stimulation.
Stimulation Time	0	Time of exogenous IFN stimulation. A negative value indicates pre-treatment with IFN.
Results to Plot	Virions	Changes the parameter plotted on the GUI plot. All active simulations will automatically be plotted by changing this.
Other Output		Other output options for Results to Plot.
Start	N/A	Performs the simulation.
Update Plot	N/A	Refreshes the GUI plot. Updates plot with new Other Output.
Log Scale Y	1 (on)	Sets the GUI plot Y axis to logarithmic scale.
Show Quantile Lines	1 (on)	Shows/hides upper and lower quantile lines.
Show All Plots	N/A	Opens 7 figures per active simulation with plots that show all output species.
Clear Data/Plot	N/A	Deletes all active simulations and clears the GUI plot.
Export Plot	N/A	Saves the GUI plot to file.
Export Results	N/A	For each active simulation, the following tables are saved to file: Parameter Output, Median Results, Upper Quantile, Lower Quantile, Mean Results, and Sensitivity Analysis (if applicable).

Note: Ten unique colors are pre-loaded, and the colors will repeat if more simulations are overlaid.

- To examine the effect of the host antiviral response, toggle the Antiviral Response switch off and click Start. For simulations at an MOI of 5, the median virion concentration is increased by toggling off the antiviral response (Figure 3A, green compared to blue). Next, change the Results to Plot dropdown to Interferon Stimulated Proteins and uncheck the Log Scale Y button. The lack of interferon stimulated proteins confirms that the antiviral response is disabled (Figure 3B, green).
- The model can also simulate the effect of adding exogenous IFN. To perform a simulation with exogenous IFN added 6 h after infection, toggle the IFN Stimulation switch on, and enter "6" in the Stimulation Time field. Next, toggle the Antiviral Response switch back on and click Start. To plot +ssRNA instead of virion concentration, change the Results to Plot dropdown to Total +ssRNA and check the Log Scale Y button. Uncheck the Show Quantile Lines button to reduce crowding and make the plot easier to interpret (Figure 4, cyan compared to blue).

Note: Entering a negative value in the Stimulation Time field simulates IFN pretreatment of cells prior to infection.

Note: The Antiviral Response switch must be turned on for the exogenous IFN to stimulate an antiviral response. Otherwise, the model does not incorporate the exogenous IFN. The model simulation is the same with and without exogenous IFN if the Antiviral Response is off.

- To save the main figure plot as it is currently displayed, press the Export Plot button. To export another plot with a different species-of-interest, change the Results to Plot dropdown then press the Export Plot button again. Press the Export Results button to save all of the underlying data. Export Results saves 5 tables per active simulation (Median, Mean, Upper Quantile, Lower Quantile, and Run Parameters).

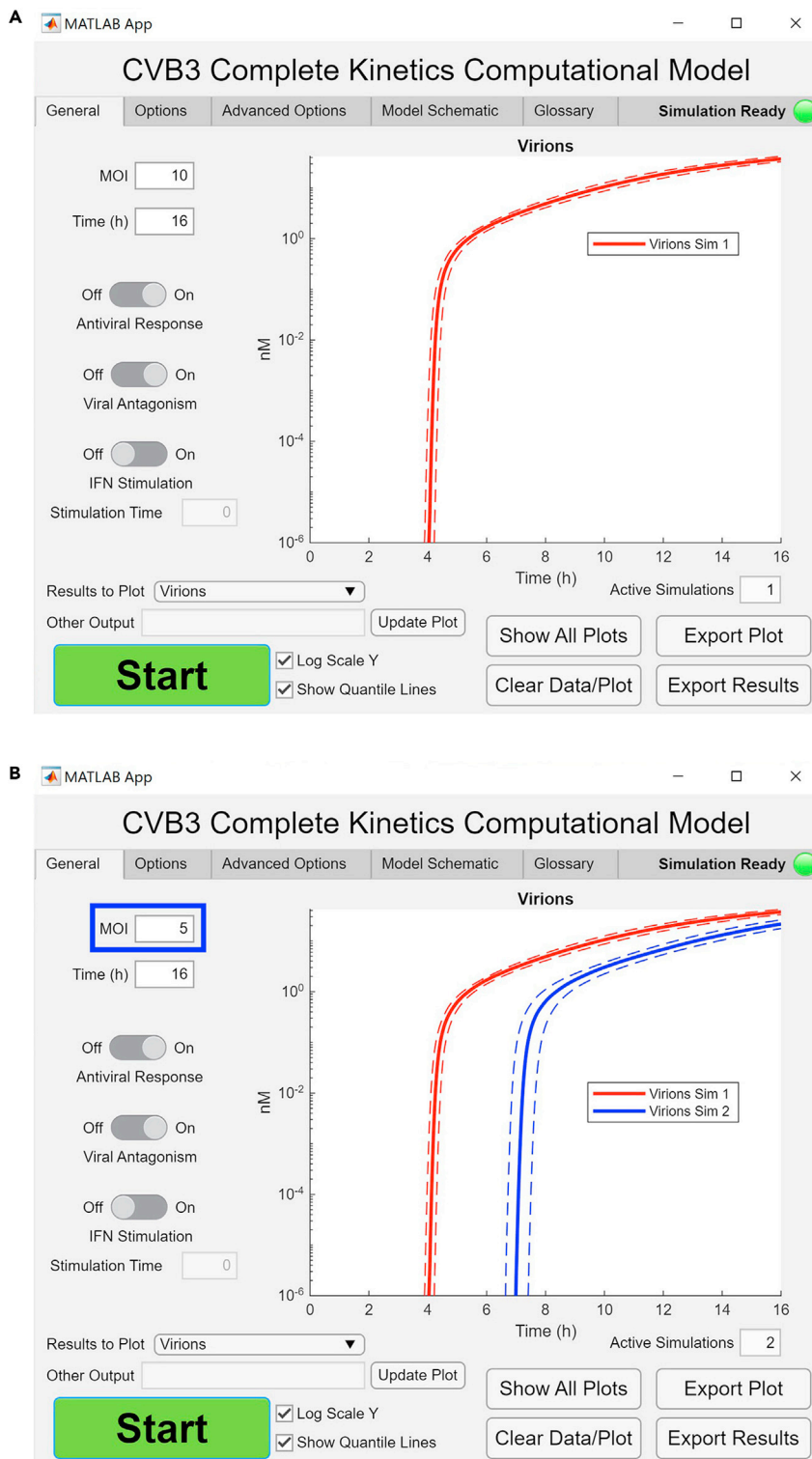


Figure 2. Demonstration of a single simulation and sequential simulations

(A) The graphical output is the result of running the model with default settings.

(B) The graphical output of a second simulation (blue) using MOI 5 is overlaid on the first simulation (red). The blue box highlights changes made for the simulation. The solid lines show the median number of virions over time, and the dotted lines show the upper (95%) and lower (5%) quantiles.

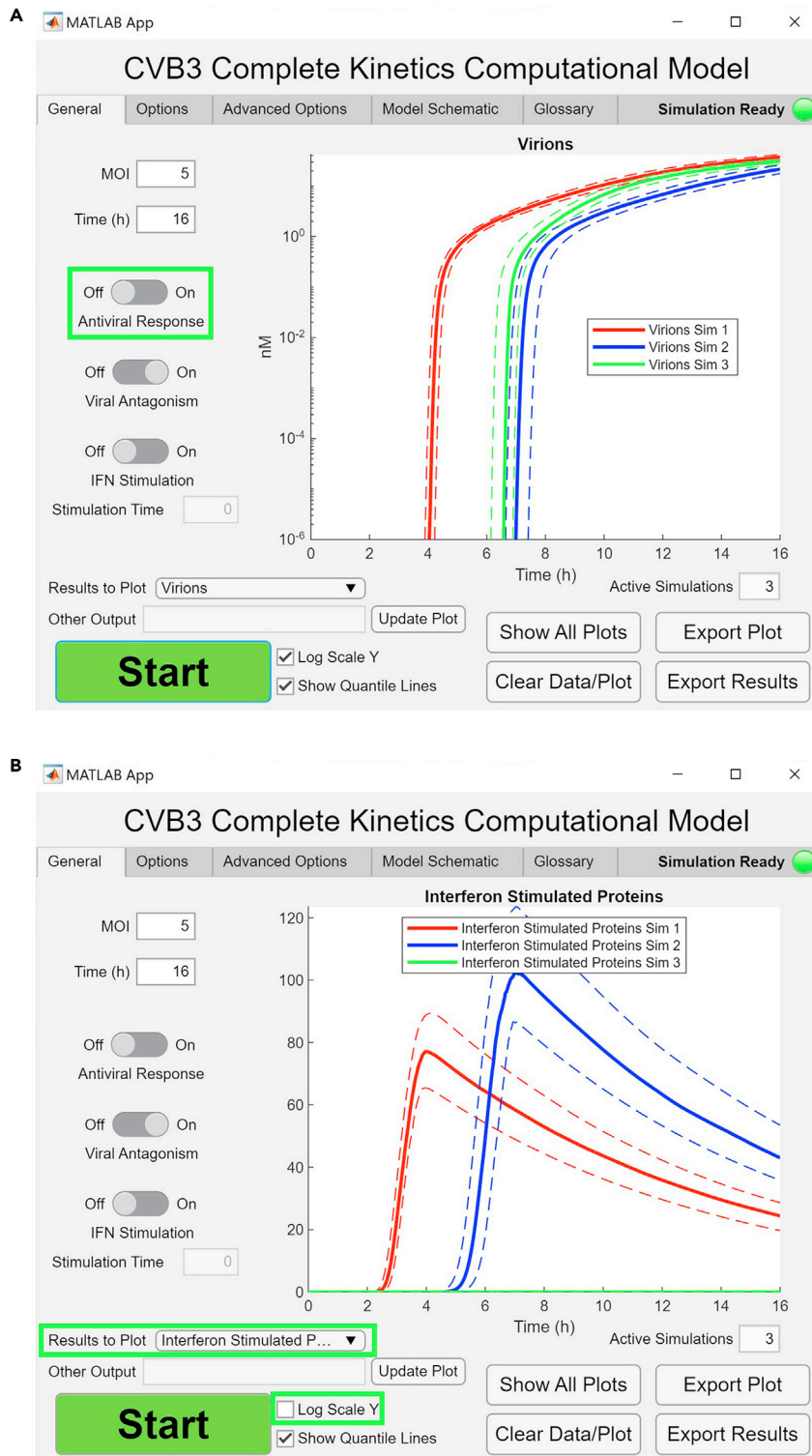


Figure 3. Turning off the antiviral response and plotting a different species

The graphical output is the result of running a third simulation (green) after the simulations shown in Figure 2 (red and blue). The third simulation was run at an MOI of 5 with the cellular antiviral response turned off.

(A) Virion production increases without an antiviral response.

(B) No interferon stimulated proteins are produced when the antiviral response is turned off. The green box highlights changes made for the simulation. The solid lines show the median number of virions over time, and the dotted lines show the upper (95%) and lower (5%) quantiles.

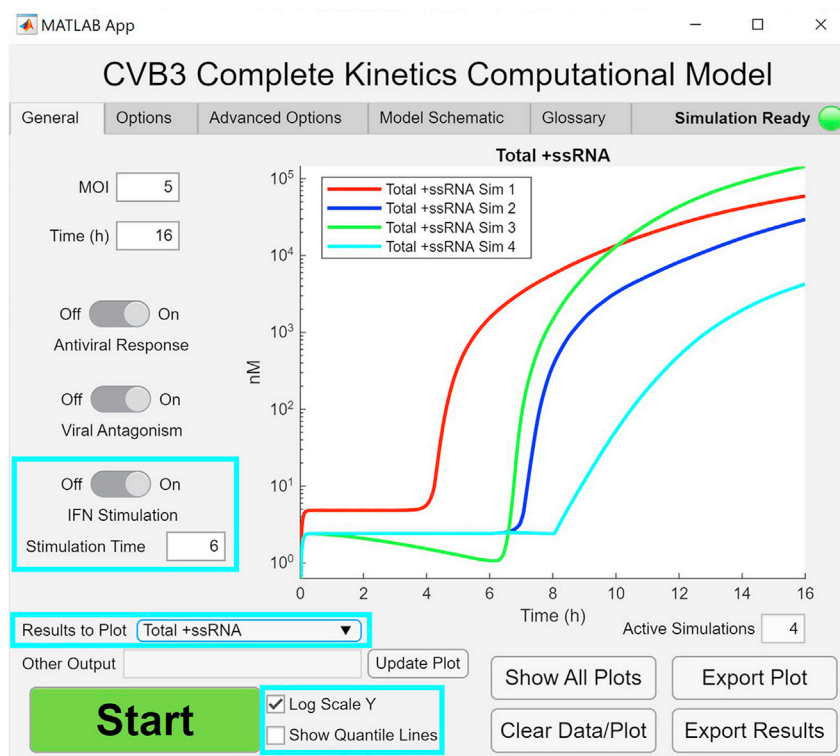


Figure 4. Adding exogenous IFN stimulation

The graphical output is the result of running a fourth simulation (cyan) after the simulation shown in Figure 3. The fourth simulation was run at an MOI of 5 with exogenous IFN stimulation starting at 6 h post infection and the host antiviral response on. The plot shows total positive-sense single-stranded RNA (+ssRNA) on a logscale y-axis with the quantile lines turned off. The cyan boxes highlight changes made for the simulation. The solid lines show the median number of virions over time.

Note: The Run Parameters table is used as a reference for the settings that went into each simulation.

Note: To view many species without changing Results to Plot, press the Show All Plots button. Show All Plots will plot all 59 output species grouped into 7 windows per simulation rather than overlaying the data from multiple simulations. For example, clicking Show All Plots after two simulations will result in 14 windows. These plots can also be saved.

- In addition to the parameters modified in the above steps, all other parameters in the General tab (Table 1), Options tab (Table 2), or Advanced Options tab (Table 3) can also be edited. Descriptions of all fields and their default values are listed in Tables 1, 2, and 3.

Note: All parameters can be reset to default values by clicking the Reset all Parameters to Default Values button in the Advanced Options tab. Resetting to default values does not clear the plot, remove active simulations, change simulation mode, or enable/disable sensitivity analysis.

Note: Simulations that result in a median virion concentration greater than 326 nM will be truncated at the time point that 326 nM is reached. See the [quantification and statistical analysis](#) for the rationale and calculation of the threshold.

Population mode

⌚ Timing: 1–2 min per simulation

Table 2. Options tab input options

Input name	Default value	Notes
ISG concentration for RNA degradation	5	EC50_RNAdeg—the half-effective concentration in nM of interferon-stimulated genes for viral RNA degradation.
Viral protease concentration for viral detector degradation	0.001	EC50_DetectorDeg—the half-effective concentration in nM of viral protease for viral detector degradation.
ISG concentration for viral protease degradation	20	EC50_Protease—the half-effective concentration in nM of interferon-stimulated genes for viral protease degradation.
ISG concentration for inhibition of viral translation	10	EC50_Translate—the half-effective concentration in nM of interferon-stimulated genes for viral translation inhibition.
Simulation Mode	Single Cell	Toggles between a Poisson distribution (population) or using a single integer (single cell) for MOI. Quantile lines are disabled in population mode. Toggling this switch deletes all active simulations and clears the GUI plot.
Sensitivity Analysis	0 (off)	Indicates if a sensitivity analysis should be simulated. During a sensitivity analysis the run count is set to one and quantile lines are disabled. Toggling this switch deletes all active simulations and clears the GUI plot.
Output species	Plus RNA	Changes Sensitivity Analysis output.
Other output		Other output options for Sensitivity Analysis. Multiple species may be entered (separated by commas, with no spaces) and the output will be the sum of the entered species.
Scaling factor	10	Factor to scale parameters by during sensitivity analysis.
Max orders of magnitude to scale	2	Orders of magnitude to scale (in both directions) during sensitivity analysis.
Number of runs	100	Indicates the number of runs in a simulation. Runs are not identical, as parameters are stochastically sampled from independent log-normal distributions.
Upper quantile	0.95	The upper quantile of run values to be plotted for indicated output species.
Lower quantile	0.05	The lower quantile of run values to be plotted for indicated output species.
Coefficient of variation	0.05	The coefficient of variation (CV) of the log-normal distribution from which the parameters are stochastically determined. A larger value indicates greater parameter variability between runs. Higher CV values can cause model instability and increase the likelihood of errors. When set at the CV maximum (0.2), approximately one error will occur every 700–800 runs.

(Simulation time may vary with computer processing speed and non-default parameter sets.)

The default simulation mode is to model the infection of a single cell with the number of virions specified in the MOI box. The model is also able to simulate infection of an entire population of cells with the discrete number of virions for each run taken from a Poisson distribution around a mean specified by the MOI, as occurs in an actual infection (Ellis and Delbrück, 1939). In population mode, each run involves a different number of virions; the mean and median for output species are plotted as the population-wide average without quantile lines.

7. Click Reset all Parameters to Default Values under the Advanced Options tab to undo all changes made thus far and reset the model. Under the Options tab, turn the Simulation Mode knob from Single Cell to Population, which will clear all active simulations. Under the General tab, click Start. Both the median and the mean virion concentration are plotted without quantile lines (Figure 5).

Note: Quantile lines are disabled in population mode. No other GUI functionality is lost. Like in single cell mode, all other model parameters can be modified, and results can be exported.

Table 3. Advanced options tab input options

Input name	Default value	Notes
DAF	4.4e+04	Initial number of unbound cell surface DAF molecules.
CAR	5.5e+06	Initial number of unbound cell surface CAR molecules.
Ribosomes	1e+05	Initial number of host cell ribosomes.
Cell volume	3700	Total volume of the cell in μm^3 . This must be greater than nucleus volume.
Nucleus volume	960	Total volume of the cell nucleus in μm^3 .
Cell surface area	2200	Total outer cell surface area in μm^2 .
VRO surface area	120	Surface area of the average viral replication organelle (VRO) in μm^2 .
CVB3 translocation to tight junction	60	k_{transloc} —the rate of DAF translocation into and out of the tight junction in h^{-1} . This rate determines the speed at which CVB3 is delivered by DAF to bind CAR in the tight junction. The default is set at maximum delivery to CAR in ~ 15 min.
CVB3 internalization	0.5	k_{internal} —the rate of internalization of the CVB3 bound to CAR $^{-1}$. This also results in downregulation of CAR at the cell surface. The default is set to achieve maximum internalization within ~ 20 min.
Endosome escape	420	$k_{\text{endo_escape}}$ —the rate of CVB3 +ssRNA release from internalized endosomes into the cytoplasm in h^{-1} . Default is set at ~ 70 particles released per 10 min.
Viral polymerase rate	180	Rate of transcription by the viral polymerase for production of + and –ssRNA, in kilobase pairs per hour (kb/h).
Host polymerase rate for ISGs	170	$k_{\text{transcribeISG}}$ —the rate of transcription for host ISGs after viral detection or interferon stimulation, in kilobase pairs per hour (kb/h).
3Dpol molecules required to form VRO	25	Number of 3D ^{pol} molecules required to form a membrane-bound VRO and accelerate viral replication.
Replication complex formation rate: +RNA	0.36	$k_{\text{R_Ip_Form}}$ —the rate of viral +ssRNA binding to 3D ^{pol} in a VRO to form a replication complex in $\text{nM}^{-1} \text{h}^{-1}$.
Replication complex formation rate: -RNA	0.36	$k_{\text{R_In_Form}}$ —the rate of viral –ssRNA binding to 3D ^{pol} in a VRO to form a replication complex in $\text{nM}^{-1} \text{h}^{-1}$.
Translation rate	3.5	The translation rate of CVB3 +ssRNA into viral polyprotein, in codon/s.
Host ribosome cleavage rate	1158	$k_{\text{cat_cleave}}$ —the rate of host ribosome (EIF4G) cleavage by viral proteases in h^{-1} . The cleavage reaction shifts host ribosomes to viral ribosomes.
Protease Km for host ribosomes	960e+03	$K_{\text{m_cleave}}$ —the Km value, indicating the affinity of viral protease for host ribosome (EIF4G) in nM. The cleavage reaction shifts host ribosomes to viral ribosomes.
Translation complex formation rate	25/Polysome size	$k_{\text{T_c_Form}}$ —the rate of viral +ssRNA binding to ribosomes to form translation complexes in $\text{nM}^{-1} \text{h}^{-1}$. Polysome size is set at 2.5.
Pentamer assembly rate	3.6	The capsid:capsid (with or without viral RNA) and capsid:RNA rate of association in $\text{nM}^{-1} \text{h}^{-1}$. This parameter simultaneously changes k_{1f_cap} , k_{2f_cap} , and $k_{\text{RNACapBind}}$.
Pentamer disassembly rate	3600	The capsid:capsid (with or without viral RNA) and capsid:RNA rate of dissociation in h^{-1} . This parameter simultaneously changes k_{1b_cap} , k_{2b_cap} , and $k_{\text{RNACapUnbind}}$.
+ssRNA degradation rate	0.23	The basal rate of positive-strand viral RNA degradation in h^{-1} . This parameter simultaneously changes $u_{\text{P_cyt}}$ and $u_{\text{P_VRO}}$.
-ssRNA degradation rate	0.23	The basal rate of negative-strand viral RNA degradation in h^{-1} . This parameter simultaneously changes $u_{\text{N_cyt}}$ and $u_{\text{N_VRO}}$.
Translation complex degradation rate	0	$u_{\text{T_c}}$ —the degradation rate for +ssRNA in translation complexes in h^{-1} . The default is set to zero under the assumption that Poly(rC) binding proteins protect the viral RNA during polysome formation.
Viral protein degradation rate	0.05	The basal rate of viral 3D ^{pol} , protease, and capsid degradation in h^{-1} . This parameter simultaneously changes $u_{\text{VirProt_cyt}}$, $u_{\text{VirProt_VRO}}$, $u_{\text{cap_cyt}}$, and $u_{\text{cap_VRO}}$.
ISG degradation rate	0.1	u_{ISG} —the rate of ISG protein degradation by the host cell in h^{-1} .

Sensitivity analysis

⌚ Timing: 2–4 min

(Simulation time may vary with computer processing speed and non-default parameter sets.)

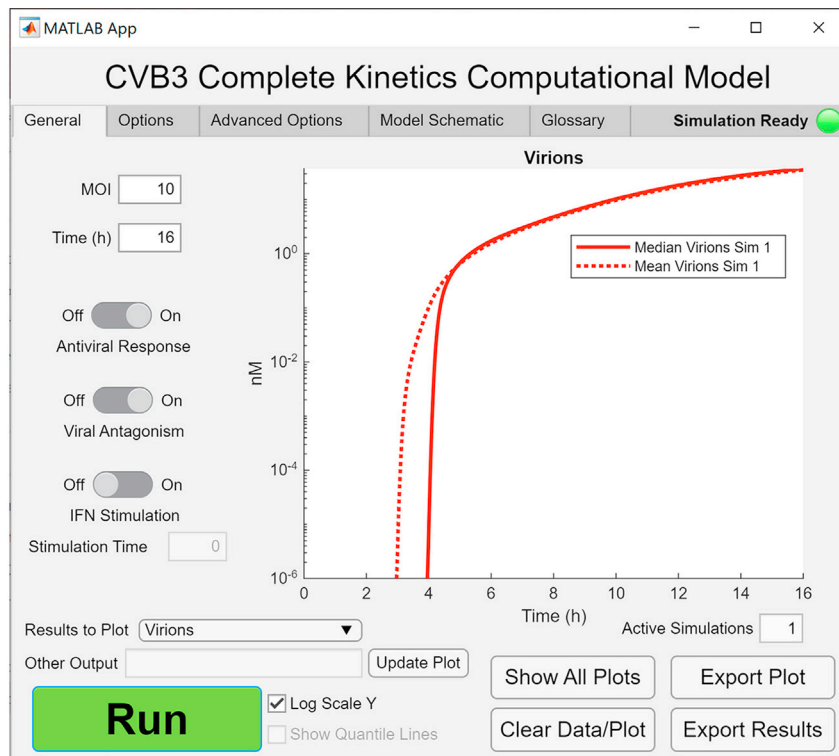


Figure 5. A simulation in population mode

The graphical output is the result of running a simulation in Population mode instead of Single Cell mode. In Population mode, the number of viral particles in each run is randomly selected from a Poisson distribution with the mean set at the input MOI. The plot shows the mean (dotted line) and the median (solid line) virions over time.

The CVB3 computational model can run a sensitivity analysis. The sensitivity analysis shows changes in a user-defined output species when parameters are individually scaled up or down. Each parameter is scaled by the value in the Scaling factor box to a maximum/minimum set by the Max orders of magnitude to scale box. The coefficient of variation among runs is set to 0 so that the parameter being scaled is the only change between iterations. As a result, quantile lines are also disabled. The sensitivity analysis output is displayed as a heatmap showing the fold change in the output species-of-interest at the simulation endpoint.

- Click Reset all Parameters to Default Values under the Advanced Options tab to reset the model. Under the Options tab, check the Sensitivity analysis box, which will clear all active simulations. Under the General tab, click Start. The default sensitivity analysis shows changes in positive-sense CVB3 RNA (+ssRNA) (Figure 6).

Note: A sensitivity analysis can be run in either single cell mode (as in Figure 6) or population mode.

Note: The sensitivity analysis output can either be a single species or the sum of any number of output species. Multiple species can be selected by selecting Other in the Output species dropdown menu and entering the output species to sum in the Other output box, separated by commas (no spaces).

Note: Similar to running the model with the sensitivity analysis disabled, any model parameter can be modified, and the results can be exported. An additional table containing the results of the sensitivity analysis will also be exported.

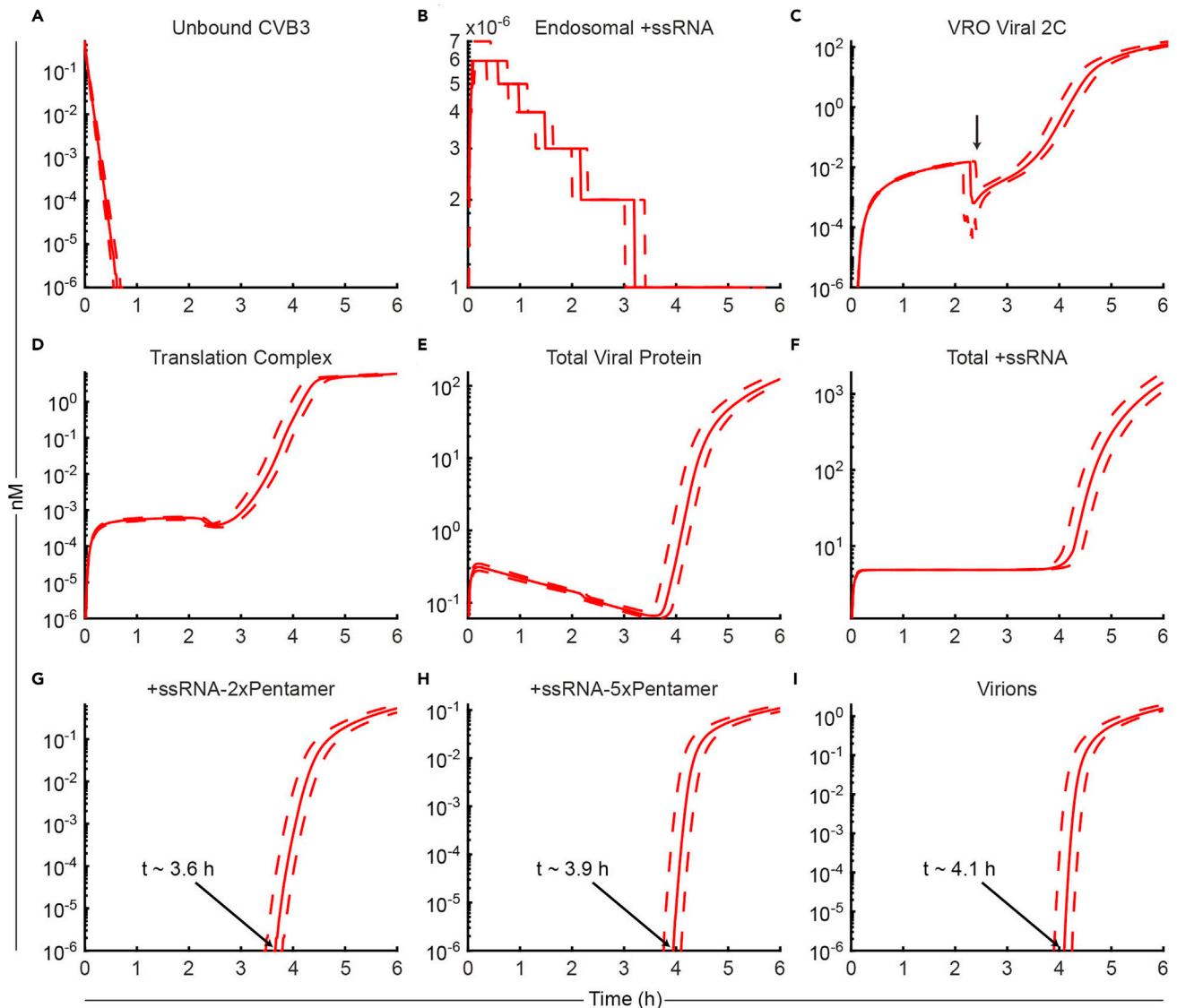


Figure 7. Model GUI outputs for the selected species during a viral life cycle

The simulation was performed for 6 h with all other parameters set to their default values.

(A and B) Species from the delivery module decrease early in the viral life cycle. The concentration of Endosomal +ssRNA (B) is low due to the transient nature of the species.

(C–F) Species from the replication module increase over the course of viral infection. Once VROs are formed (C, arrow), the number of translation complexes rapidly increases (D), which causes an explosive increase in viral proteins (E) that can then replicate the viral genome (F).

(G–I) Species from the encapsidation module appear in a time-dependent manner. Early RNA–pentamer species (G) appear before intermediate RNA–pentamer species (H), which appear earlier than filled virions (I). +ssRNA: positive-sense single-stranded viral RNA. VRO: viral replication organelle. +ssRNA-*n*xPentamer: one +ssRNA bound to *n* associated pentamers. The simulations were run in the CVB3 model GUI, and the plots were collected using the Export Plot button then processed in Adobe Illustrator. The solid lines show the median number of virions over time, and the dotted lines show the upper (95%) and lower (5%) quantiles.

et al., 2015; Wang et al., 2020). To model polymerase inhibition, four simulations were run with iterative threefold decreases in the viral polymerase rate (180–6.67 kb/h). As expected, decreasing the polymerase rate did not affect the rate of viral entry, indicated by the level of endosomal +ssRNA (Figure 8A). However, there was a threshold effect for the +ssRNA and viral protein, with substantial decreases observed at the lowest polymerase rates (Figures 8B and 8C). The decreased protein abundances arise from less +ssRNA template available for translation, rather than a decrease in

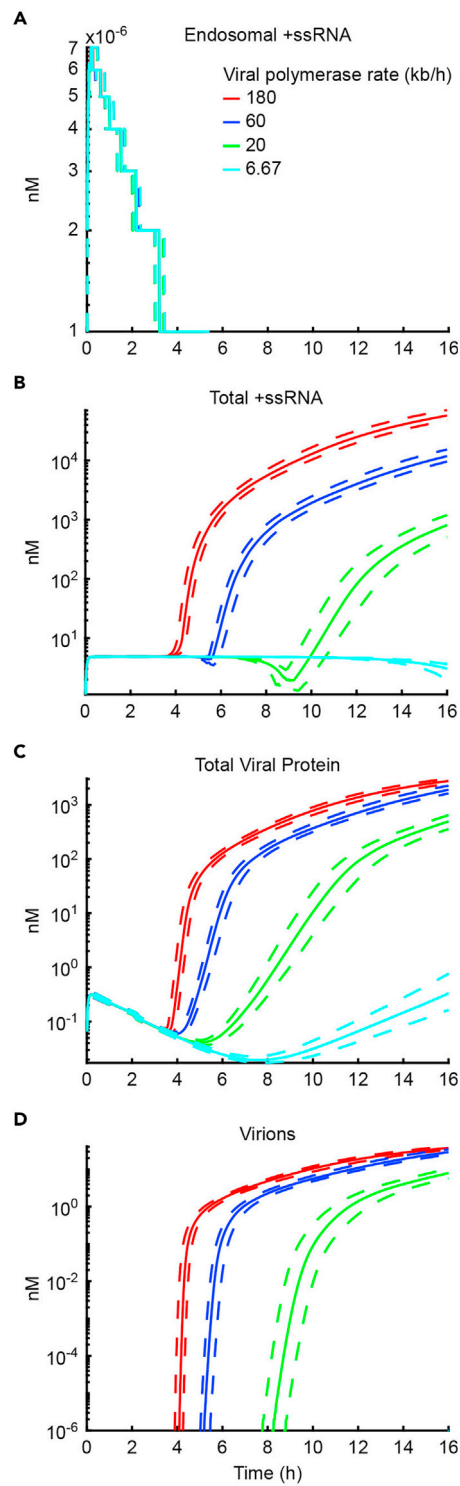


Figure 8. Model GUI outputs for selected species identify a threshold viral polymerase rate for productive infection

Four simulations were performed with iterative threefold decreases in the viral polymerase rate (180–6.67 kb/h). All other simulation parameters were set to default.

(A) Viral entry—measured by endosomal positive-sense single-strand viral RNA (+ssRNA)—is unaffected by the viral polymerase rate. The concentration of Endosomal +ssRNA is low due to the transient nature of the species.

Figure 8. Continued

(B and C) Species in the replication module are sensitive to the viral polymerase rate. The decrease in Total +ssRNA (B) decreases the amount of available template for translation of viral proteins (C). (D) A threshold viral polymerase rate for productive infection. No virion production is observed for the lowest viral polymerase rate (6.67 kb/h), presumably due to the lack of +ssRNA (B) and capsid protein (C). The simulations were run in the CVB3 model GUI, and the plots were collected using the Export Plot button then processed in Adobe Illustrator. The solid lines show the median number of virions over time, and the dotted lines show the upper (95%) and lower (5%) quantiles.

the translation rate itself. Similarly, the number of live virions was dramatically decreased at the lower polymerase rates (Figure 8D).

In some instances, it may be more appropriate to perform simulations in population mode rather than single cell mode. One such scenario is at a low MOI. See the [population mode](#) section for additional information on the differences in the MOI between single cell mode and population mode. To demonstrate, a simulation was run in both single cell mode and population mode at an MOI of 2. In single cell mode, two virions are not sufficient to cause a productive infection for the given time frame (Figure 9A). In population mode, the median agrees with the single cell median, as expected, because the population median represents a single run at an MOI approximately equal to the MOI used in single cell mode (Figure 9B). However, some runs in population mode simulate infection for more than two virions since the number of virions in each run is pulled from a Poisson distribution. Therefore, some runs yield a productive infection that is quantifiable when looking at the population-wide average (Figure 9B). When using the CVB3 model to conduct experiments, it can be worthwhile to compare single cell mode and population mode to determine if run-to-run variation in MOI affects the experimental outcome.

QUANTIFICATION AND STATISTICAL ANALYSIS

The CVB3 complete kinetics computational model performs repeated runs for a simulation (default: Number of runs = 100). For each run in a simulation, parameter values are drawn from a log-normal distribution. Each log-normal distribution is parameter-specific and centered about the user-defined

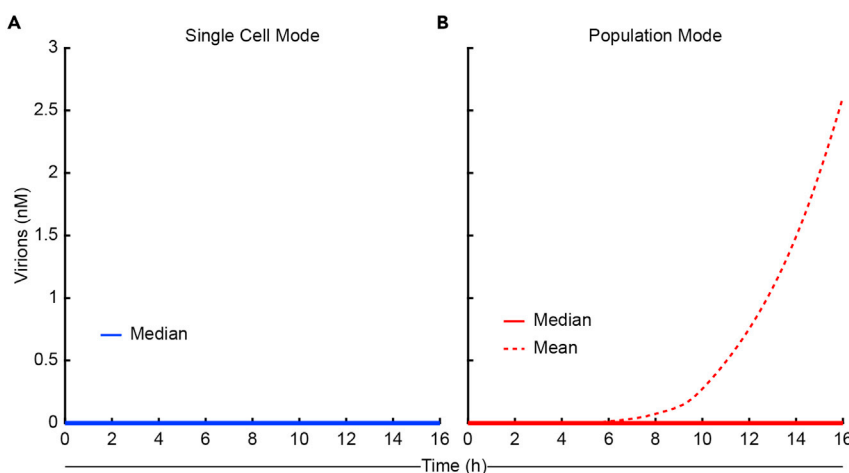


Figure 9. Model GUI outputs comparing single cell mode to population mode at a low MOI

The simulation was performed for 16 h with MOI set at 2 in either Single Cell mode (A) or Population mode (B) with all other parameters set to their default values.

(A) At a low MOI, a single cell does not yield a productive infection (solid blue).

(B) In a population of cells, the median response is the same as in a single cell (solid red), while the mean virion production of the entire population yields a quantifiable response (dotted red). The simulations were run in the CVB3 model GUI, and the plots were collected using the Export Plot button then processed in Adobe Illustrator.

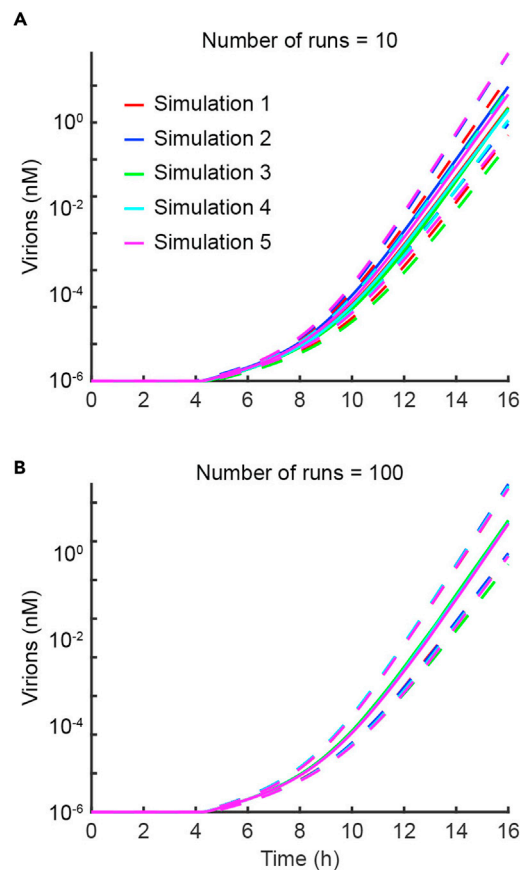


Figure 10. Model GUI outputs comparing serial simulations with different run counts

All other parameters were set to their default values.

(A) Five simulations with Number of runs set to 10.

(B) Five simulations with Number of runs set to 100. The simulations were run in the CVB3 model GUI, and the plots were collected using the Export Plot button then processed in Adobe Illustrator. The solid lines show the median number of virions over time, and the dotted lines show the upper (95%) and lower (5%) quantiles.

value of the parameter. The default values are derived from the literature or experiments. The spread of each log-normal distribution is determined by a user-defined coefficient of variation (default: 5%). Drawing from log-normal distributions causes run-to-run variability in output species. Therefore, the data are summarized as the median value \pm confidence interval (default: 90%; determined by the 5th and 95th percentiles of the runs for a simulation). As a result, changing the run count, coefficient of variation, or quantile values will all change the output graphs. For instance, increasing the coefficient of variation will increase the spread of log-normally drawn parameter values and increase the range of output species. In the model GUI, the coefficient of variation is limited to a maximum of 20%, as the model becomes unstable at higher values. Due to differences in the pseudorandom number generation seed, performing the same simulation twice may yield slightly different results. We have found that there is moderate variability between simulations with a per-simulation run count of 10 (Figure 10A). By contrast, a run count of 100 is sufficient to limit simulation-to-simulation variability (Figure 10B). However, some simulation conditions may be intrinsically more variable and require an increased run count to obtain stable predictions.

Understanding how the model generates uncertainty is also important to draw parallels between model outputs and biology. In the model, it is assumed that every parameter varies a relatively small amount between runs, with all parameters varying to a similar degree. However, in biology these

assumptions are not necessarily true, where some features between cells may vary to a great degree (such as protein abundances), while others presumably change very little (such as protein-protein binding rates). Population mode accounts for biological uncertainty in how many live virions infect a cell, but it cannot account for all cell-to-cell variability. As a result, the uncertainty generated by the model should be used to evaluate the variation in model runs and not be interpreted as biological variation.

At the end of the replication cycle, enterovirus virions are released through host-cell bursting. Cell bursting is not encoded in the command-line version of the model, so viral species increase to unrealistic concentrations in some simulation conditions (e.g., over long periods of time). To increase user-friendliness in the GUI, we hardcoded a threshold for the maximum virion concentration based on estimates of the maximum plaque-forming units (PFU) produced by poliovirus in a cell of similar volume to the “cell volume” parameter in the GUI (Dunnebacke and Reaume, 1958). We converted from PFU to the number of particles using the reported particle-to-PFU ratio (Lopacinski et al., 2021). Then, we converted the number of particles to a concentration using Equation 1:

$$\text{concentration (nM)} = \frac{x}{N_A} \times \frac{1}{V_{\text{cyto}}} \times 10^{24} \quad (\text{Equation 1})$$

where x = the number of particles; N_A = Avogadro’s number, $6.022 \times 10^{23} \text{ mol}^{-1}$; V_{cyto} = cytoplasm volume in μm^3 , found by subtracting the nuclear volume from the cell volume; and 10^{24} is the conversion factor to obtain concentration in nanomolar (nM).

The calculated maximum virion concentration for the default cytoplasmic volume in the GUI is 326 nM. Beyond the threshold, cell lysis is assumed, and the output data is truncated. Further, we calculated the virion concentration at which cell lysis would be likely to occur based on the median PFU produced by poliovirus (Dunnebacke and Reaume, 1958). The median concentration is 88 nM. Simulated virion outputs greater than 88 nM trigger a warning message that reminds the user that concentrations beyond 88 nM are likely to cause cell lysis.

LIMITATIONS

All computational models require assumptions and simplifications, and the CVB3 complete kinetics model is no exception. Assumptions are a necessary trade-off between model accuracy and functionality. Overly complex models often have reduced functionality—increased computational expense and difficult interpretability—and obtain only a marginal increase in molecular detail. Simpler models can be more useful.

One frequently used simplification is lumping a group of reactions or species into a single parameter. For the CVB3 model, three portions of the life cycle were lumped. First, several CVB3 proteins were lumped together. CVB3 encodes two different proteases (2A and 3C), which each have different host cell substrates. In the complete kinetics model, $2A^{\text{pro}}$ and $3C^{\text{pro}}$ are lumped into a single species. Similarly, $3D^{\text{pol}}$ was grouped with the 3AB protein since they strongly interact (Xiang et al., 1998), and three capsid subunits (VP0, VP1, and VP3) were grouped into a single capsid protomer. The remaining hydrophobic proteins were lumped into a single protein ($2C^{\text{ATPase}}$). Second, capsid formation was greatly simplified. Biologically, the binding affinity for subsequent pentamers increases as more capsid pentamers associate and more contact points are available for the next pentamer. Rather than model every possible capsid-forming configuration and its associated binding affinity, the CVB3 computational model neglects geometry and uses the median affinity for every step of capsid formation. Third, the innate immune response is collapsed into a single equation. The full cellular antiviral response to CVB3 consists of a sensor (MDA5) transducing a signal through MAVS and Interferon Regulatory Factor 3 (IRF3) to induce the expression of IFN. IFN is secreted and stimulates the infected cell in an autocrine fashion through the IFN α/β receptor and JAK/STAT pathway to induce the expression of ISGs (Sadler and Williams, 2008). In the CVB3 model, the entire innate immunity pathway is simplified into a Hill equation, which is fit to ISG expression

data from a cardiomyocyte cell line. Despite the simplifications, the complete kinetics model quantitatively captures crucial pieces of the CVB3 life cycle in molecular detail.

The CVB3 complete kinetics model uses mass action to simulate the concentrations of different species in the cell over time. As a result, all output values for measurable species are reported as nanomolar concentrations. Interpreting concentration is difficult for some species, such as virions, which are typically measured in plaque-forming units per milliliter (PFU/mL). The number of PFUs in a viral stock is not reflective of the total number of virions in the stock. In a viral stock, some virions may lack RNA (empty), and others may be incapable of forming plaques (defective). The model generates empty virions, but it does not capture the generation of defective virions. Furthermore, host-cell variations determine whether a live virion will form a plaque in culture (Goh et al., 2016). Therefore, model outputs and experimental results involving virions are separate from one another: 1) The complete kinetics model does not account for defective particles, meaning it overestimates the concentration of virions capable of forming a plaque; and 2) Experiments yielding data in PFU/mL do not give any information on the total number of assembled virions. Furthermore, the output virion concentration from the model is the intracellular concentration, since there is no cell lysis component encoded. Instead of a lysis component, a maximum intracellular virion concentration is set in the model, and data beyond the maximum is truncated. Taken together, virion outputs from the model should be treated in relative terms, where the relative change between treatments is more informative than the absolute numbers.

Parameter values in the CVB3 model are set to literature- or experimentally-derived estimates. However, some parameter values are unavailable in the CVB3 literature but are available for closely related viruses; therefore, some default values were derived from literature on other enteroviruses, such as poliovirus. Using parameters derived from other viruses introduces a degree of uncertainty to the model. However, the viral proteins encoded in the model are shared among enteroviruses, and their viral replication cycles are highly conserved. Thus, the parameters likely deviate very little from the actual CVB3 values, lending confidence to the real-world utility of the model.

The CVB3 complete kinetics model GUI functions identically to the command line-based version of the model and contains many adjustable parameters. However, the GUI has some limitations compared to the command line-based CVB3 model. Specifically, 26 parameters are editable using the command line but not in the GUI. These parameters and the rationale for not including them in the GUI are listed in Table 4. Additionally, the underlying equations behind the model are rigidly set in the GUI version but can be customized in the command line-based model. This limitation balances user-friendliness and customizability. The GUI version of the model was created to enable any user to run simulations without needing MATLAB installed or knowledge of how to code in MATLAB. If users require the greater range of functions associated with the command line-based model, it is available for download (<https://github.com/JanesLab/CompleteKinetics-CVB3>).

TROUBLESHOOTING

Problem 1

The simulation program will not open.

Potential solution

The simulation program will not open if MATLAB Runtime (or the full version MATLAB) is not installed in the correct location. Double check that version 9.10 (R2021a) of MATLAB Runtime is installed. On MacOS systems, MATLAB Runtime must be in the Applications directory in a folder named "MATLAB" (install the graphical user interface (GUI), steps 3e and 4a – Figure 1). On Windows systems, MATLAB Runtime must be correctly installed and present in the Add or remove programs list (install the graphical user interface (GUI), step 2a). If MATLAB Runtime cannot be located, download and install the latest version for the appropriate operating system (<https://www.mathworks.com/products/compiler/mcr/index.html>). Alternatively, if the simulation will not open after completing the automatic installation instructions, it may help to instead follow the manual installation instructions.

Table 4. Parameters from the complete kinetic model that are excluded from the CVB3 model GUI

Name	Description	Notes
RibAvail	Initial concentration of ribosomes available to the virus and not host.	This parameter is dependent on the total number of ribosomes.
RibUnavail	Initial concentration of ribosomes available to the host and not virus.	This parameter is dependent on the total number of ribosomes.
CVB3diameter	Diameter of a CVB3 virion.	
Pol3DLength	Length of the CVB3 3D polymerase.	
ISGbpLength	Length of ISG15 in kilobase pairs, which is used to determine the rate of ISG formation.	Changing the host polymerase rate for ISGs would have a similar effect as changing this parameter.
RiboActive	Proportion of all ribosomes that are currently active in the cell.	Changing the total number of ribosomes would have a similar effect as changing this parameter.
PolysomeSize	Average number of ribosomes bound to a single mRNA during translation.	
CVB3genomelength	Length of the CVB3 genome in nucleotides.	
CVB3polyprolength	Length of the CVB3 polyprotein in amino acids.	
VROVolConv	Volume conversion to model the concentrating effect of VROs.	Parameter is dependent on cell volume, nucleus volume, and VRO surface area.
k_on_DAF	Association rate between CVB3 and DAF.	Changing the DAF concentration would have a similar effect as changing this parameter.
k_off_DAF	Dissociation rate between CVB3 and DAF.	
k_on_CAR	Association rate between CVB3 and CAR.	Changing the CAR concentration would have a similar effect as changing this parameter.
k_off_CAR	Dissociation rate between CVB3 and CAR.	
k_Translate	Overall rate of CVB3 translation.	Calculated from the translation rate, polysome size, and polyprotein length.
k_N_Transcript	Rate of CVB3 negative-strand synthesis.	Calculated from the viral polymerase rate and CVB3 genome length.
k_P_Transcript	Rate of CVB3 positive-strand synthesis.	Calculated from the viral polymerase rate and CVB3 genome length.
k_P_on	Rate of CVB3 positive-strand translocation to the VRO.	
k_P_off	Rate of CVB3 positive-strand translocation out of the VRO.	
k_N_on	Rate of CVB3 negative-strand translocation to the VRO.	
k_N_off	Rate of CVB3 negative-strand translocation out of the VRO.	
k_Pentamer_on	Rate of CVB3 capsid pentamer translocation to the VRO.	
k_Pentamer_off	Rate of CVB3 capsid pentamer translocation out of the VRO.	
n_Hill	Hill coefficient used to model viral detection by the host cell.	Fit to ISG expression data from cardiomyocytes.
kD_Hill	Dissociation constant used in the Hill equation to model viral detection by the host cell.	Fit to ISG expression data from cardiomyocytes.
ISGForm	Formation rate of ISG proteins.	Changing the host polymerase rate for ISGs would have a similar effect as changing this parameter.

Problem 2

The simulation progress bar freezes.

Potential solution

The simulation will freeze under three circumstances:

First, if the Time (h) box is set too high, the amount of data stored may cause an out of memory error. To avoid this error, reduce the Time (h) value. In test simulations, a Time (h) value of over 50,000 led to model freezing. (The maximum value depends on the specifications of the computer used to run the simulation.) Similarly, exporting data and clearing all active simulations may help if the GUI is running slowly due to a lack of memory ([basic operations](#), step 5).

Second, the underlying ordinary differential equation solver encounters model parameters that are difficult to simulate. Avoid exploring input parameter combinations that lead to model freezing. For example, increasing the +ssRNA degradation rate to 5 and the Time (h) value to 96 can lead to model freezing ([basic operations](#), step 6).

Third, incompatible input parameter values are entered into the model. Input parameters have been constrained in the GUI to avoid incompatibilities with the model. These incompatibilities are the reason why the coefficient of variation has been constrained to a maximum of 20%. However, if simulations are run with the coefficient of variation at or near 20%, freezing due to parameter incompatibility remains a possibility ([quantification and statistical analysis](#)). Reduce the coefficient of variation to avoid this problem.

Unfortunately, the only solution when the model GUI is completely frozen is to close the GUI and re-open it. All data from previously active simulations will be lost. To prevent this error from occurring in the future, avoid circumstances that lead to conditions prone to freezing.

Problem 3

No data appear in main GUI plot after running a simulation.

Potential solution

Check that the Simulation Ready lamp has turned from red to green. The green light indicates that the program has completely finished running and plotting the simulation. If there are still no data in the main GUI plot, uncheck the Log Scale Y checkbox ([basic operations](#), step 3). It is possible that the output species from the simulation is at a concentration of zero throughout the simulation and therefore cannot be plotted when the Y axis is set at a log scale. Using a linear scale, the data will appear as a line along the base of the plot if the species concentration is at zero ([Figure 3B](#), green). Alternatively, press the Reset all Parameters to Default Values button (in the Advanced Options tab) and run a default simulation to restore data plotted in the main GUI plot.

Problem 4

The model displays the same output between two simulations when a parameter is changed, even though different output is expected.

Potential solution

Depending on the simulation, one of two things may be happening:

First, if the simulation involves an exogenous interferon stimulation (i.e., IFN Stimulation Time field is on), ensure that the Antiviral Response switch is toggled on ([basic operations](#), step 4). If the Antiviral Response switch is off, IFN stimulation will not affect the model simulation. If toggling the Antiviral Response switch on still does not achieve the expected result, see the next potential solution.

Second, it is possible that the current set of parameter values overshadows the effect of an additional change. Click the Reset all Parameters to Default Values button (in the Advanced Options tab) to restore the default values and run a baseline simulation to ensure the model is functioning ([basic operations](#), step 6). Then, alter the parameter(s)-of-interest, and perform another simulation.

If the expected output is still not achieved, the chosen parameter values may represent a scenario in which there is truly no change to the system or the differences are so small that the tolerances in the model do not reflect any change.

Problem 5

The traces in the graphical output are erratic or irregularly shaped.

Potential solution

Graphical traces will be irregularly shaped when the ordinary differential equation solver encounters very small parameter values that cause discontinuities in the solution. Irregularities are observed under two conditions:

First, irregularities occur when plotting output species at extremely low concentrations (e.g., Endosomal +ssRNA, [expected outcomes](#) – [Figure 7B](#)). It should be noted that low concentration-based discontinuities are artifactual and should not be interpreted as discrete jumps in species abundance.

Second, discrete changes in species concentrations, such as during the VRO transition, cause artifacts in the traces ([expected outcomes](#) – [Figure 7C](#)). This problem can be mitigated by exercising caution when interpreting graphs with irregular traces. Alternatively, changing altered parameters closer to their default values can improve erratic model outputs.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Kevin Janes (kjanes@virginia.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The graphical user interface used during this protocol is available at <https://github.com/JanesLab/CompleteKinetics-CVB3-GUI>.

ACKNOWLEDGMENTS

We thank Lionel Jensen, Quinten Kieser, and Ryley McClelland for beta-testing the GUI. This work was supported by the David and Lucile Packard Foundation (2009-34710, K.A.J.), a gift from the Salameone Family Foundation (K.A.J.), the Human Frontier Science Program (LT000469/2021-L, C.D.G.), and training grant support from the National Institutes of Health (T32-HL007284, A.J.S.).

AUTHOR CONTRIBUTIONS

Conceptualization, A.J.S., C.D.G., and K.A.J.; methodology, A.J.S., C.D.G., and K.A.J.; software, A.J.S. and C.D.G.; validation, A.J.S. and C.D.G.; investigation, A.J.S. and C.D.G.; resources, A.J.S., C.D.G., and K.A.J.; data curation, A.J.S. and C.D.G.; writing — original draft, A.J.S. and C.D.G.; writing — review & editing, A.J.S., C.D.G., and K.A.J.; visualization, A.J.S. and C.D.G.; supervision, K.A.J.; project administration, K.A.J.; funding acquisition, A.J.S., C.D.G., and K.A.J.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

- Andrejeva, J., Childs, K.S., Young, D.F., Carlos, T.S., Stock, N., Goodbourn, S., and Randall, R.E. (2004). The V proteins of paramyxoviruses bind the IFN-inducible RNA helicase, mda-5, and inhibit its activation of the IFN-beta promoter. *Proc. Natl. Acad. Sci. U S A* *101*, 17264–17269. <https://doi.org/10.1073/pnas.0407639101>.
- Bergelson, J.M., Cunningham, J.A., Droguett, G., Kurt-Jones, E.A., Krithivas, A., Hong, J.S., Horwitz, M.S., Crowell, R.L., and Finberg, R.W. (1997). Isolation of a common receptor for Coxsackie B viruses and adenoviruses 2 and 5. *Science* *275*, 1320–1323. <https://doi.org/10.1126/science.275.5304.1320>.
- Bruggeman, F.J., and Westerhoff, H.V. (2007). The nature of systems biology. *Trends Microbiol.* *15*, 45–50. <https://doi.org/10.1016/j.tim.2006.11.003>.
- Coyne, C.B., and Bergelson, J.M. (2006). Virus-induced Abl and Fyn kinase signals permit coxsackievirus entry through epithelial tight junctions. *Cell* *124*, 119–131. <https://doi.org/10.1016/j.cell.2005.10.035>.
- Dunnebacke, T.H., and Reaume, M.B. (1958). Correlation of the yield of poliovirus with the size of isolated tissue cultured cells. *Virology* *6*, 8–13. [https://doi.org/10.1016/0042-6822\(58\)90055-2](https://doi.org/10.1016/0042-6822(58)90055-2).
- Ellis, E.L., and Delbrück, M. (1939). The growth OF bacteriophage. *J. Gen. Physiol.* *22*, 365–384. <https://doi.org/10.1085/jgp.22.3.365>.
- Etchison, D., Milburn, S.C., Ederly, I., Sonenberg, N., and Hershey, J.W. (1982). Inhibition of HeLa cell protein synthesis following poliovirus infection correlates with the proteolysis of a 220,000-dalton polypeptide associated with eucaryotic initiation factor 3 and a cap binding protein complex. *J. Biol. Chem.* *257*, 14806–14810.
- Feng, Q., Langereis, M.A., Lork, M., Nguyen, M., Hato, S.V., Lanke, K., Emdad, L., Bhoopathi, P., Fisher, P.B., Lloyd, R.E., and van Kuppeveld, F.J.M. (2014). Enterovirus 2Apro targets MDA5 and MAVS in infected cells. *J. Virol.* *88*, 3369–3378. <https://doi.org/10.1128/JVI.02712-13>.
- Goh, K.C.M., Tang, C.K., Norton, D.C., Gan, E.S., Tan, H.C., Sun, B., Syenina, A., Yousuf, A., Ong, X.M., Kamaraj, U.S., et al. (2016). Molecular determinants of plaque size as an indicator of dengue virus attenuation. *Sci. Rep.* *6*, 26100. <https://doi.org/10.1038/srep26100>.
- Jiang, P., Liu, Y., Ma, H.-C., Paul, A.V., and Wimmer, E. (2014). Picornavirus morphogenesis. *Microbiol. Mol. Biol. Rev.* *78*, 418–437. <https://doi.org/10.1128/MMBR.00012-14>.

- Kang, D., Gopalkrishnan, R.V., Wu, Q., Jankowsky, E., Pyle, A.M., and Fisher, P.B. (2002). mda-5: an interferon-inducible putative RNA helicase with double-stranded RNA-dependent ATPase activity and melanoma growth-suppressive properties. *Proc. Natl. Acad. Sci. U S A* 99, 637–642. <https://doi.org/10.1073/pnas.022637199>.
- Kawai, T., Takahashi, K., Sato, S., Coban, C., Kumar, H., Kato, H., Ishii, K.J., Takeuchi, O., and Akira, S. (2005). IPS-1, an adaptor triggering RIG-I- and Mda5-mediated type I interferon induction. *Nat. Immunol.* 6, 981–988. <https://doi.org/10.1038/ni1243>.
- Kempf, B.J., and Barton, D.J. (2008). Poliovirus 2A(Pro) increases viral mRNA and polysome stability coordinately in time with cleavage of eIF4G. *J. Virol.* 82, 5847–5859. <https://doi.org/10.1128/JVI.01514-07>.
- Lopacinski, A.B., Sweatt, A.J., Smolko, C.M., Gray-Gaillard, E., Borgman, C.A., Shah, M., and Janes, K.A. (2021). Modeling the complete kinetics of coxsackievirus B3 reveals human determinants of host-cell feedback. *Cell Syst.* 12, 304–323.e13. <https://doi.org/10.1016/j.cels.2021.02.004>.
- Melia, C.E., Peddie, C.J., de Jong, A.W.M., Snijder, E.J., Collinson, L.M., Koster, A.J., van der Schaar, H.M., van Kuppeveld, F.J.M., and Bárcena, M. (2019). Origins of enterovirus replication organelles established by whole-cell electron microscopy. *mBio* 10, e00951–19. <https://doi.org/10.1128/mBio.00951-19>.
- Miller, S., and Krijnse-Locker, J. (2008). Modification of intracellular membrane structures for virus replication. *Nat. Rev. Microbiol.* 6, 363–374. <https://doi.org/10.1038/nrmicro1890>.
- Mukherjee, A., Morosky, S.A., Delorme-Axford, E., Dybdahl-Sissoko, N., Oberste, M.S., Wang, T., and Coyne, C.B. (2011). The coxsackievirus B 3C protease cleaves MAVS and TRIF to attenuate host type I interferon and apoptotic signaling. *PLoS Pathog.* 7, e1001311. <https://doi.org/10.1371/journal.ppat.1001311>.
- Racaniello, V. (2013). Picornaviridae: the viruses and their replication. In *Fields Virology*, D.M. Knipe and P.M. Howley, eds. (Wolters Kluwer/Lippincott Williams & Wilkins Health), pp. 452–489.
- Sadler, A.J., and Williams, B.R.G. (2008). Interferon-inducible antiviral effectors. *Nat. Rev. Immunol.* 8, 559–568. <https://doi.org/10.1038/nri2314>.
- Seth, R.B., Sun, L., Ea, C.-K., and Chen, Z.J. (2005). Identification and characterization of MAVS, a mitochondrial antiviral signaling protein that activates NF-kappaB and IRF 3. *Cell* 122, 669–682. <https://doi.org/10.1016/j.cell.2005.08.012>.
- Shafren, D.R., Bates, R.C., Agrez, M.V., Herd, R.L., Burns, G.F., and Barry, R.D. (1995). Coxsackieviruses B1, B3, and B5 use decay accelerating factor as a receptor for cell attachment. *J. Virol.* 69, 3873–3877. <https://doi.org/10.1128/JVI.69.6.3873-3877.1995>.
- van der Linden, L., Vives-Adrián, L., Selisko, B., Ferrer-Orta, C., Liu, X., Lanke, K., Ulferts, R., De Palma, A.M., Tanchis, F., Goris, N., et al. (2015). The RNA template channel of the RNA-dependent RNA polymerase as a target for development of antiviral therapy of multiple genera within a virus family. *PLoS Pathog.* 11, e1004733. <https://doi.org/10.1371/journal.ppat.1004733>.
- Wang, M., Li, R., Shu, B., Jing, X., Ye, H.-Q., and Gong, P. (2020). Stringent control of the RNA-dependent RNA polymerase translocation revealed by multiple intermediate structures. *Nat. Commun.* 11, 2605. <https://doi.org/10.1038/s41467-020-16234-4>.
- Xiang, W., Cuconati, A., Hope, D., Kirkegaard, K., and Wimmer, E. (1998). Complete protein linkage map of poliovirus P3 proteins: interaction of polymerase 3Dpol with VPg and with genetic variants of 3AB. *J. Virol.* 72, 6732–6741. <https://doi.org/10.1128/JVI.72.8.6732-6741.1998>.