Open science approaches to the mathematical modeling of infectious disease

Simon Frost

Microsoft and LSHTM

A large number of mathematical and computational models of infectious disease have been developed, particularly in the last fifty years. Despite this rich resource, numerical analysis of these models by third parties is complicated by many factors, including the lack of detail in how the model was developed and the numerical methods used, and the use of unmaintained software packages. Models of infectious disease can be made more findable, accessible, interoperable, and reproducible through a combination of; (a) specific formulations of the models; (b) packaging of the model and any dependencies; and (c) adding a programming interface to these models.

Construction of a compartmental model of disease transmission usually starts by writing down equations for the rates of change of each compartment, e.g. (S)usceptibles, (I)nfecteds, and (R)ecovereds. However, a more flexible approach starts with writing equations for the processes underlying the rates of change e.g. transmission and recovery, an approach widely taken e.g. in chemical kinetics models. This representation of Markovian models can be easily converted into ordinary differential equations, stochastic differential equations, or jump processes, and enables the automated construction of models for the moments and the probabilities of a given state of the stochastic system at a particular time. Many models are complex, but can be made simpler to understand if they are constructed from smaller building blocks, wired together using a causal approach, where outputs from one submodel are fed into the inputs of another model, or an acausal approach, with potentially bidirectional inputs/outputs, and where common states in the submodels are identified. Use of specific software can also make models easier to use e.g. the use of domain-specific languages to describe a model that make the computer program look more similar to the mathematical equations, or more powerful, e.g. the use of software that can use automatic differentiation to calculate gradients, allowing techniques such as local sensitivity analysis and inference using Hamiltonian Monte Carlo to be used.

For models with numerical results, the written equations are an incomplete description of the entire system. The numerical precision used, the numerical solver used (along with the associated discretization of continuous variables), and even the underlying computer package can all affect the output. In addition, software systems evolve over time, and may become unmaintained. To overcome this, a model along with all the dependencies and data required to run the model should be packaged together in order to allow a third party to exactly replicate the results of a model. One technique to do this is to use a ‘container’, a kind of lightweight virtual machine. One advantage of this approach is that it also allows running the model in cloud computing environments, allowing much bigger models and larger parameter spaces to be considered.

Even with a model running in a container, a third party is likely to want to be able to change the parameter values of the model and observe changes in the associated output. Each model may have its own formats for model inputs and outputs, making it time consuming as well as potentially error-prone to run a model. One potential remedy is to build an interface to the model through the use of Uniform Resource Identifiers (URIs), which are used to interact with web pages. Not only does this allow running of a model through a web browser or a command line, but it also allows verification of model inputs, reducing the risk that misleading outputs will arise as a consequence of inappropriate choice of input parameters. Modern computer tools allow this to be done with a few extra lines of code.

By following these open science practices, modelers can take advantage of platforms that can run models in an automated way, such that multiple models can be more easily compared. Interoperability of models means that developments e.g. in modeling transmission or in numerical approaches can be quickly and easily incorporated. By defining models in terms of their inputs and outputs, modelers can work with e.g. data scientists to build platforms that feed data to the models, without the data scientists needing to know the details of the model architecture. Through an open science approach, theoretical and applied model development can be accelerated and distributed more widely, increasing its impact in multiple areas of scientific endeavour.