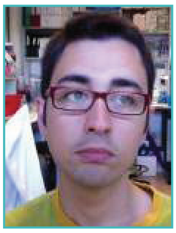
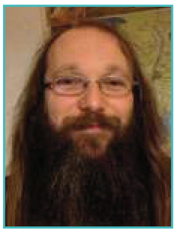
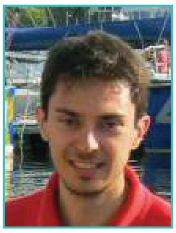
Quick Direct-method Controlled (QDC): a simulator of metabolic experiments

Davide Cangelosi[1], Salvatore Fabbiano[2], Claudio Felicioli[3], Luca Freschi[4], Roberto Marangoni[5,*]

[1]Institute Gianna Gaslini, Genova, Italy
[2]University of Salamanca, Spain
[3]Noname Research, Italy
[4]Universite' Laval, Quebec City, Canada
[5]University of Pisa, Italy
* corresponding author (marangon@di.unipi.it)

Received 10 August 2012; **Accepted** 11 Feb-2013; **Published** 19 August 2013
Competing Interests: none

ABSTRACT

Quick Direct-method Controlled (QDC) isa stochastic simulator based on the direct method version of Gillespie's Stochastic Simulation Algorithm(SSA). It has been specifically designed to simulate experiments performed on metabolic networks, when external operators can act on the system, modifying its spontaneous behaviour. Users of QDC can simulate different experimental controls:i.e.,add or remove chemical species at a given time; change the rate of a reaction at a given moment; and describe reactions with complex stoichiometrythat take place once the stoichiometric condition is verified (here called immediate reactions). Moreover, even though QDC is not designed to manage compartments, it can simulate up-take and excretion reactions. QDC represents a useful tool for the specific field of interest thanks to its computational performances and simple input language.

Introduction

Gillespies' Stochastic Simulation Algorithm (SSA) (Gillespie, 1977) is the most widely used approach to simulate the stochastic behaviour of metabolic networks, in particular when the number of molecules involved is relatively small. There are several variants of SSA: the direct-method was the first proposed, followed by many others that address different issues related to the reduction of high computational costs, or dealing with stiff systems, etc. (for a review, see Li *et al.*, 2008). Modern stochastic simulation suites offer wide libraries of different simulation methods, analysis facilities and numerical resources, and can face multi-scale systems, hybrid deterministic and stochastic representations, and many other problems that occur in systems biology. Nevertheless, during some of the studies we carried out on signalling pathways and minimal cells,we needed to extend the description of the target systemby including at least three control actions that would be possible to perform on it: a) to add/rule out a given amount of molecules of a given species at a given time; b) to change the propensity of a reaction,depending on the presence/absence of an external trigger signal; and c) to specify all-or-nothing reactions.

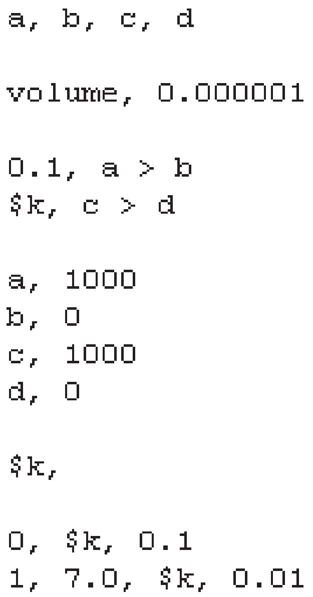
We first tried to describe such extended models using different existing simulation packages, but we encountered severe difficulties; therefore, we developed QDC specifically to match these needs in an easy and efficient way. QDC does not aim to compete with existing suites, but rather, to complement them for these specific tasks.

QDC

QDC's syntax and input file structure

QDC's core input is represented by an ASCII file written in anin-house format (see Figure 1). Supplementary file 1[1] offers a detailed description of the QDC input language and usage;more detail is presented in the README file that comes with the package download. Here, we give a brief summary.The formal definition of the whole syntax is presented in Supplementary file 2[2].

The input file is structured in sequential blocks, separated by a blank line; each block contains a different category of information: block B1 declares the names of the chemical species present in the system; B2 declares the system volume, measured in litres, that is used for computing stochastic propensities; B3 declares the reactions, using a notation very close to that of standard biochemistry. Each reaction is introduced by the deterministic reaction rate coefficient, which QDC transforms into the stochastic propensity, according to the reaction order, system volume and species concentrations. Among the different reactions, QDC allows users to declare zero-order reactions, useful to simulate the uptake of a chemical species from a generic outside. The left-hand member of the chemical equation is here represented by the operator 'NULL'. First-order reactions with the right-hand part of the equation containing only the operator NULL can be used to simulate excretion reactions. A special reaction class is constituted by the so-called immediate reactions. Theseare introduced by a dash sign instead of a coefficient, and take place immediately after the verification of the condition represented by the left-hand side of the equation. In other words, the stoichiometry noted on the left-hand side of the equation is a logical condition: once it is verified, the immediate reaction takes place, and yields the products indicated on the right-hand side. These reactions allow for complex stoichiometry, where both the reagent species number and the number of molecules per species can be greater than two. We want to remark that *immediate reactions* are not to be considered higher-order reactions,asthey do not have a kinetic law, and they happen immediately. Immediate reactions can also be used to simulate the simultaneous excretion of several chemical species, once they have reached a given threshold number of molecules: to do so, the right-hand side of the equation must contain only the 'NULL'term.



```
a, b, c, d

volume, 0.000001

0.1, a > b
$k, c > d

a, 1000
b, 0
c, 1000
d, 0

$k,

0, $k, 0.1
1, 7.0, $k, 0.01
```

Figure 1.

Immediate reactions represent the most prominent innovative feature implemented in QDC with respect to other simulation packages. Immediate reactions can be seen as infinite-propensity reactions, as, in fact, they are. Nevertheless, they avoid several implementation problems that a generic infinite-propensity reaction can have. First, immediate reactions avoid the zero-times-infinity problem that will affect a generic infinite-propensity reaction when the reagent species concentrations are nearly zero; second, immediate reactions make it no longer necessary to describe the kinetic law for reaction of order higher than two. Immediate reactions turn out to be very useful in describing all-or-nothing reactions that take place in different biological contexts. For example, the firing of a neuron, which happens when enough synaptic stimulations are received, can be modelled by using immediate reactions.

B4 contains the number of molecules assigned to all the declared chemical species that

[1] http://journal.embnet.org/index.php/embnetjournal/article/downloadSuppFile/505/505 _ supp _ 1

[2] http://journal.embnet.org/index.php/embnetjournal/article/downloadSuppFile/505/505 _ supp _ 1

are supplied to the system, along with the time at which they are supplied. When such time is set to zero, the statement denotes the initial number of molecules for the declared species.

B5 and B6(optional) are concerned with the eventual presence of controls exerted on kinetic coefficients that are not constant during the simulation, but change according to some trigger signal: in particular,B5declares the name of all dynamic coefficients used in the simulation (their name begins with a dollar sign); B6 declares the starting instant of the value change and the new value assumed by the variable.

QDC's core

QDC's core, written in C++, implements Gillespie's direct method (Gillespie, 1977) to simulate the time course of a biochemical system. In order to avoid possible violations of the algorithm correctness, QDC does not use any approximation. Given a metabolic system (and eventual actions performed on it) described in an ASCII input file, QDC parses it into C++ source files, compiles and executes them. This procedure has been designed to specialise the source file on the given model, thus allowing one to fully exploit the compiler optimisations. Thanks to this metacompilation approach, which, to the best of our knowledge,is used for the first time in a metabolic network simulator, QDC offers very good computational performances.

QDC's output

QDC outputs four files: the first contains the time course of the number of molecules of each simulated metabolite; the second contains the counters of each metabolic reaction firing; the third contains the time course of the propensity for each reaction, and the fourth is a log-file of the computation. Files concerning reaction-fire counts and time-course of propensities complement the information contained in the first, helping users to reconstruct more accurate views of the dynamics of the system. For instance, if a given chemical species counts 0 molecules in the reagent file, one cannot conclude that it does not exist: it may be that the genesisreactions are slow with respect to the consuming reactions – then the balance of that species at any sampling point will result in 0, but the species is dynamically existent and gives its contribution to the system evolution. This can be detected by inspecting the reaction-counter file, where one can assess that both genesisreactions and consuming reactions have effectively fired. By examining the propensities time-course file, one can understand which reactionscommandthe highest importance in the system at a given time, therebyrevealingthe evolution of hub-pathways over time.

SBML Import/Export

The QDC package also contains two applications (import_sbml.py and export_sbml.py) that provide the Import(I)/Export(E) from/to SBML[3] level 2, ver. 4, thanks to the libSBML v.4.0[4] libraries. Of course, such I/E is limited to the expressions and the statements that both the languages (of SBML and QDC) support. The SBML I/E can also be managed via the Graphical User Interface (GUI). Supplementary file 3[5] gives a detailed description in SBML of the three main control events managed by QDC.

QDC's GUI

QDC's GUI has been developed to giveusers an immediate visualisation of the simulated system behaviour. The GUI is basic and easy to use: it has been developed in Python v.2.6 and uses the PyQt libraries[6] to manage the interface's elements. This choice confers good portability to-QDC's GUI, as it has been tested on different Linux distributions (Fedora, Ubuntu, *etc.*) and on Mac OS X.

A benchmark test

A very basic benchmark test was run, based on a comparison between QDC and three widely used simulators –StochKit (Kierzek, 2002), Dizzy (Ramsey *et al.*, 2005) and BetaWorkBench (Dematté *et al.*, 2008) – which offer an implementation of Gillespie's direct method. To compare the relative efficiency in implementing Gillespie's algorithm, it is not correct to compare the runtime required by each simulator in simulating the same input model: this kind of measure, in fact, is influenced by other implementation characteristics (disk usage, memory allocation, etc.). Instead, we ran several simulations of the same biochemical model (one strictly similar to that presented in Figure 1, but without control variables)by using the same computer, by varying the simulated time, and measuringthe machine time required to per-

[3] www.sbml.org
[4] http://sbml.org/Software/libSBML
[5] http://journal.embnet.org/index.php/embnetjournal/article/downloadSuppFile/505/505_supp_1
[6] http://www.riverbankcomputing.co.uk/software/pyqt

form the task. We built, for each simulator, a curve of dependence of the actual elapsed time vs. the simulated time.There is a region where these curves are quasi-linear (when the simulation requires a number of operations significantly greater than those necessary to launch the program, but not so high as to require the computer to start swapping). We determined, by using a linear regression, the slope of these lines, which represents the coefficient linking the elapsed time to the simulation time. Their average values were as follows: StochKit = 7.25 10^{-3}, Dizzy = 5.8 10^{-3}, BetaWorkBench = 4.09 10^{-3} and QDC= 2.75 10^{-3} (these average values were computed on 1,000 runs; the standard error was less than 10% of the value). We remark that this test does not represent a study on QDC's complexity, nor a proper screening of QDC performance: it represents only an indication that QDC's efficiency can be assumed comparable with that of other freely available simulators.

Availability

QDC is freely available, under the GPL v3 license, through the SourceForge platform[7].

Acknowledgments

The authors thank Davide Chiarugi and Lorenzo Lazzerini for the validation tests they executed,and Roberto Grossi and Pierpaolo Degano for their critical reading of the manuscript, and the formal syntax language in Supplementary file 2.

References

Dematté L, Priami C, Romanel A (2008) The Beta Workbench: a computational tool to study the dynamics of biological systems. *Briefings in Bioinformatics* **9**, 437-449.doi: 10.1093/bib/bbn023.

Gillespie DT (1977) Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem.* **81**, 2340-2361. doi: 10.1021/j100540a008.

Kierzek AM(2002) STOCKS: Stochastic Kinetic Simulations of biochemical systems with Gillespie algorithm. *Bioinformatics* **18**, 470-481. doi: 10.1093/bioinformatics/18.3.470.

Li H, Cao Y, Petzold LR, Gillespie DT(2008) Algorithms and Software for Stochastic Simulation of Biochemical Reacting Systems. *Biotechnol. Prog.* **24**, 56-61.doi: 10.1093/bib/bbn050.

Ramsey S, Orrell D, Bolouri H(2005) DIZZY: Stochastic simulation of large-scale genetics regulatory networks. *J. Bioinf. Comp. Biol.* **3**, 415-436. doi: 10.1142/S0219720005001132.

[7] http://sourceforge.net/projects/gillespie-qdc