Institut für Informatik
Praktische Informatik und Bioinformatik
Prof. Dr. Ralf Zimmer
Dr. Evi Berchtold, Markus Joppich, Felix Offensperger, Jewgenija Zhdanovich

Corona/Online Winter Term 2020/21
Computational Systems Biology

Assignments 2020-2
Detailed analysis of small-scale dynamical systems

Working period: Three weeks (1.12.-22.12.2020)
Hand-in anytime or in any exercise class
Please hand-in only reproducible results, answers, figures, tables, simulations, ...

This project develops tools and methods for the analysis and visualisation of (small) dynamic systems. Many biological systems are specified by more or less detailed interconnected systems, whose structure is given and whose dynamical behaviour needs to be understood. The project considers various specifications of ”reaction systems”, either via equations, Petri Nets, or XML-based description languages, their transformation into executable models and appropriate execution engines (solvers, simulators). Visualisations should be produced, which elucidate the behaviour/function of the model. Ideally, visualisations could be used in scientific publications, i.e. time series of the respective variables of the model.

The solution of such models (and the visualisation of the results) depends on 1.) the specification of the model (i.e. are all rate constants and dependencies specified such that a (system of) ODE(s) can be fully stated and solved via integration, or is a partial model given such that only a crude partial simulation is possible) 2.) modelling choices (is a continuous time-resolved behaviour wanted, or the states in predefined discrete steps, the final states in a number of discrete conditions, the final outcome after convergence, the steady-state-behaviour, attractors, oscillations, ...) 3.) the methodological framework: ODE solvers, numerical solutions, stochastic simulation, Petri net semantics.

Often only a sketch (from a textbook or paper is given). Then the system according to criteria 1.-3. has to be derived from the sketch. Parameters need to be specified, or learned from some data, or extracted from appropriate databases (maybe different versions for diverse contexts), or modifications need to be made to ready-made available systems models.

There are many databases around which collect prepared models in various formats (e.g. BioModels, https://www.ebi.ac.uk/biomodels). There are also many tools, which allow to specify reaction systems, both general (e.g. matlab, octave) or special (e.g. Gepasi, Copasi, CellDesigner, ...) modelling tools.

This project is about systematic modelling with various tools, own scripts and publicly available systems. Reaction systems are specified in a simple reaction equations, `A + X -> B + Y`, or graphical sketches, e.g. Project 1, Task 1 (b). These need to be parsed, complemented, analysed, and visualised.

The goal of the project is to learn about the standard models and modelling tools, both by re-implementing some tools yourself and by learning about and making available standard reaction systems modelling frameworks.

The projects will be extended e.g. by systematic stochastic simulations and Petri net modules, but also towards large-scale, genome-wide networks and high-throughput data in subsequent tasks and projects.

Implementing small-scale dynamical reaction systems

Task 1 Specifications of (biomolecular) reaction systems

(a) Review options for specifying reaction systems, both structure and dynamics (kinetics).

Write parsers (input routines) for (one or several of) the following options

(b) Reaction equations

Write a parser for equation based specifications of a set of reactions R, such as:

$$R = \{R_1, \ldots, R_N \ \mid \ R_i : s_1, \ldots, s_n \rightarrow p_1, \ldots p_m\}$$

where the system S consists of a set of N reactions R and each reaction consumes n substrates s_i and produces m products p_j.

This defines the "structure" of the reaction system, more information associated to "$\rightarrow$" and maybe also to the substrates and products is needed for a full-scale dynamical simulation/solution of the system (reversible vs irreversible, rates, kinetic functions, parameters).

(c) Graphical Sketches

Given a graph in a standard notation (see Project 1, Task 1 (b)), e.g. represented as a list of nodes and edges or similar. What is the disadvantage of simple graph models?

(d) Graphical Sketches as Petri Nets:

Given a Petri Net in a standard notation (see Project 1, Task 1 (b)), e.g. represented as list of nodes/places P and transitions T and/or a incidence matrix I with $|P|$ rows and $|T|$ columns ($R_t : s_1, s_2 \rightarrow p_1, p_2$ or graphically $s_1, s_2 \rightarrow t \rightarrow p_1, p_2$, then $I(s_1, t) = I(s_2, t) = 1$ and $I(p_1, t) = I(p_2, t) = -1$).

(e) SBML

The Systems Biology Markup Language is a standard format to specify biomolecular models. Write a basic tool to read-in the "structure" of SBML models as in (b)-(d).

(f) Write a "converter" between the various formats, e.g. reading in an SBML model and output the reaction equations or a reaction graph / Petri net.

Task 2 Semantics of (biomolecular) reaction systems

Add a semantics to the structure of a reaction systems, which makes it a dynamical system. E.g. assign (python) functions to the "→" such that the system can be executed, simulated, or solved.

One option is to convert the reactions in a system of ODEs , which can be integrated with your python tools (similar to Project 1).

Visualise the resulting time courses for given initial values.

Task 3 ODEs (Standard Kinetics) Implement a toolbox of standard kinetics for your python ODE-solver. Allow for specifying the kinetics type for edges/transitions.

Task 4 Parameter scanning and sampling

The behaviour of dynamical models depends on a number of parameters.

(a) Scan: To get an overview, a a systematic grid search (scan) of certain regions of the parameters can be performed: select a parameter combination, run the model with theses parameters, record some output variable(s) and visualise the dependency of the selected output variables on the values of the selected parameters. Beware of 'combinatorial explosion'!

(b) Sampling: Due to the huge number of combinations of parameter values, often only a sampling of parameter space is possible, in order to get at least a broad overview of behaviours. Consider a simple sampling approach for sampling and repeat (a).

(c) Plot "phase diagrams" to illustrate the dynamical behaviour of two selected output variables.

Task 5 Numerical solution of ODEs

(a) Review algorithms/methods for the numerical integration/solution of ODEs (see lecture). Implement such a procedures, e.g. a Runge-Kutta method.

(b) Analyse numerical stability and runtime.

(c) Compare the results with a standard ODE solver.

Standard toolboxes for reaction systems

Task 6 Copasi

The Copasi tool (Pedro Mendes et al, http://copasi.org/) is a powerful app and framework for the tasks defined above - and many more.

Install Copasi and learn about its features. Analyse and describe a couple of models.

Which features can be re-implemented in your python implementations from Tasks 1-3 above (reports, Latex output, various input and output formats)?

Task 7 Octave

Octave (https://www.gnu.org/software/octave/) is a public domain scientific programming language, very similar to and compatible with the commercial tool matlab (https://www.mathworks.com). It provides a general purpose scripting language with very powerful solving and visualisations features. Moreover, Biomodels often provide a direct specification via executable octave/matlab scripts.

Install octave and learn how to solve modelling tasks via octave.

Task 8 SBML, BioModels

SBML (https://sbml.org) is "a free and open interchange format for computer models of biological processes. SBML is useful for models of metabolism, cell signaling, and more" and constitutes a whole toolset of libraries, tools, editors, simulators, etc. including APIs and language bindings for several programming environments including python.

The BioModels resource (https://www.ebi.ac.uk/biomodels/) at EMBL-EBI provides thousands of curated and non-curated models, mainly based on ODEs from a variety of research, papers, and pathways. BioModels aims to provide a comprehensive collection of "existing literature-based physiologically and pharmaceutically relevant mechanistic models in standard formats. Our mission is to provide the systems modelling community with reproducible, high-quality, freely-accessible models published in the scientific literature."

Install SBML and make use of SBML tools and the access to BioModels.