The python packages CompartmentalSystems, LAPM and bgc-md for the analysis of compartmental dynamical systems

Verónika Ceballos-Núñez[1], Holger Metzler[1], Markus Müller[1], and Carlos A. Sierra[1]

[1]Max Planck for Biogeochemistry, Hans-Knöll-Str. 10, 07745 Jena, Germany

June 29, 2018

1 People and Contributions

The presented Software is the achievement of genuine teamwork. To be able to compete in the categories of group leaders, postdocs and Ph.D. students we give a simplified summary.

Name	**Position**	**Contributions**
Verónika Ceballos-Núñez	Ph.D. student	database population report generation abstract vegetation model
Holger Metzler	Ph.D. student	algorithm development, implementation, testing, database population report generation
Markus Müller	postdoc, developer	technical lead, algorithm development, implementation, testing, report generation, refactoring, infrastructure
Carlos Sierra	group-leader	abstract models, database population, outside collaboration, organization, funding

2 Introduction

Compartmental systems are a large class of dynamical systems used in many different scientific applications where mass balance considerations are included. The presented software framework consists of three open-source python packages that serve to represent, classify, and analyze compartmental systems of ordinary differential equations (see Definition 1 in Appendix A), with special emphasis on the computation of age and transit time densities of the contents in each reservoir and the entire system. The packages do not have a graphical user interface, and are meant to be used together with other open source software e.g. jupyter.

2.1 CompartmentalSystems package

`https://github.com/MPIBGC-TEE/CompartmentalSystems`
The package allows the computation of ages and transit times for nonlinear, non-autonomous well mixed compartmental systems. A brief summary is found under the link above. A more detailed description of the concepts and their application is given in **?**. A preprint is attached `./PNAS.pdf`. An example jupyter notebook that shows how to use the package can be found here: `http://compartmentalsystems.readthedocs.io/en/latest/_downloads/nonl_gcm_3p.html`

2.2 LAPM package

`https://github.com/MPIBGC-TEE/LAPM`
LAPM stands for **L**inear **A**utonomous **P**ool **M**odels. It provides the class LinearAutonomousPoolModels that allows the *symbolic* computation of transit time, system age, and pool ages. Since transit time and system age are phase-type distributed, the computations of them rely on properties of this distribution. They are treated in a separate module. LAPM can be seen as a symbolic companion to CompartmentalSystems for the special case of linear autonomous compartmental systems. Details about the mathematical derivation of the formulas implemented in this package can be found in **?**.

2.3 The biogeochemical model data base

`https://github.com/MPIBGC-TEE/bgc-md`
This package presents a specific application for dynamical systems that represent element cycling in ecosystems. Short summaries are found under the link above or on our group website `https://www.bgc-jena.mpg.de/TEE/software/bgc-md/`. The package provides:

1. Collections of yaml files, each encoding a published carbon cycling model.
2. The code to produce (at the moment still static) html for user specified queries, or can be used in a jupyter notebook.

The software simplifies the abstract description of element cycling models and introduces a yaml format to store them. It uses a symbolic mathematical representation based on sympy, and also allows immediate numerical computations using scipy and our own packages CompartmentalSystems and LAPM. Its main use is to make complex models immediately available for further investigation e.g. in jupyter notebooks, but it also provides a command-line tool to build html reports about single models or sets. Users can create their own templates and yaml files. We are developing a web front end that makes it possible to run the whole database as a website with a graphical JavaScript based user interface. The GUI further simplifies the creation of the database entries.

3 Installation, documentation and demonstrations

3.1 Installation

For the purpose of the contest it suffices to install bgc-md because it will draw in the other packages as dependencies. Detailed instructions are found here. The installation is tested for linux systems. Since the only dependencies are pandoc and python3 it should be possible to be used on Windows or Mac OS X, but we do not maintain installation instructions for these operating systems. Instead, we provide docker images that are ready to run. I have to upload them to docker so that they can be installed easily by the users and explain the installation command

3.2 Documentation

CompartmentalSystems has a complete documentation as well as LAPM. The bgc-md package has the documentation of the command-line tool which is available after installation `render -h` and is otherwise mostly documented by the report templates, which are constantly tested automatically.

3.3 Demonstration

To see how bgc-md can be used in conjunction with the other packages look at the following jupyter notebook. To be able to see the command-line tool in action you have to install the package and make sure that your virtual environment is activated. To check that you have done so successfully, type `render -h` which should show the help. In the folder `bgc-md/docs/SiamSoftwareContest/` we provided a small shell script that contains an example call. Then run `./demo1.sh` (which will take some time). Afterwards you can look at the result with a web browser. `firefox output/TECO/CompleteSingleModelReport.html`. To see an overview report run: `./demo2.sh` and look at: `output22/Website.html`

I have to update the template for the model links in the website

A Compartmental systems

For the purpose of understanding the applications of the software it suffices to interpret all state variables as contents of reservoirs. In the following we treat here reservoir, pool, and compartmental systems as synonymous, and use relevant definitions from **?**, and references therein. The interpretation easiest to imagine is content measured in units of mass stored in a reservoir defined by its spatial boundaries. However, reservoirs as well as contents can be much more abstract.

Definition 1 (Compartmental system) *Let $F_{i,j}$ be the flux from pool j to pool i, for all $i, j \neq i \in \{1 \dots n\}$, $F_{i,0} = I_i$ define the influx to pool i and $F_{0,i} = O_i$ the outflux from pool i.*
If $F_{i,j}(\vec{C}, t) \geq 0$, for all $i, j \in \{0 \dots n\}$ and

$$C_j = 0 \implies F_{i,j}(\vec{C}, t) = 0 \tag{1}$$

we call the ODE system:

$$\dot{C}_i = \sum_{i=0.i \neq j} (-F_{j,i}(\vec{C}, t) + F_{i,j}(\vec{C}, t)) \quad \forall i \tag{2}$$

compartmental.

Remarks:
The conditions guarantee mass balance, non-negative fluxes, and ensure that there cannot be any flux out of an empty pool. For non-negative initial values $\vec{C}_0 \in \mathbb{R}^{+^n}$ the pool contents stay non-negative for $t \geq t_0$.

References

1. usage examples for bgc-md that work
2. automated tests that make sure that all the models in data/tested
3. rebuild the docker images and update the installation instructions
4. update the website template

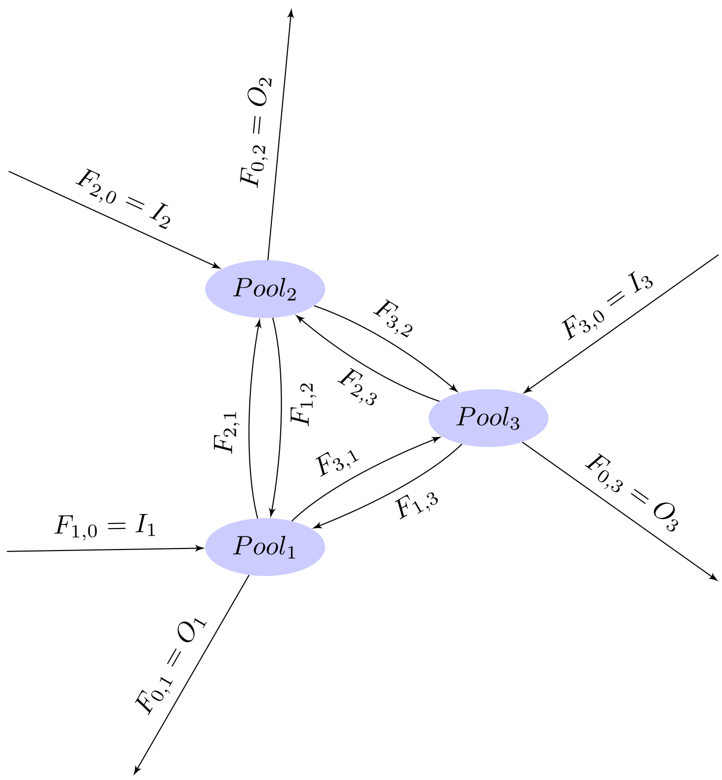

Figure 1: A compartmental model with three pools and all possible connections between the pools and the environment. Note that the flux between pool_k and pool_l is considered a property of the *pipeline*, meaning the flux from pool_k into the pipeline is the same as the flux out of the pipeline into pool_k. This ensures mass balance for all fluxes and pools and also for the model as a whole, and makes this the prototype or normal form of mass balanced models with internal connections. Every model that can be drawn in this form is mass balanced. Examples are not limited to networks of pipelines and reservoirs of fluids. Instead of a fluid the total amount of a chemical element in different chemical substances (represented by the pools) can be described by it.