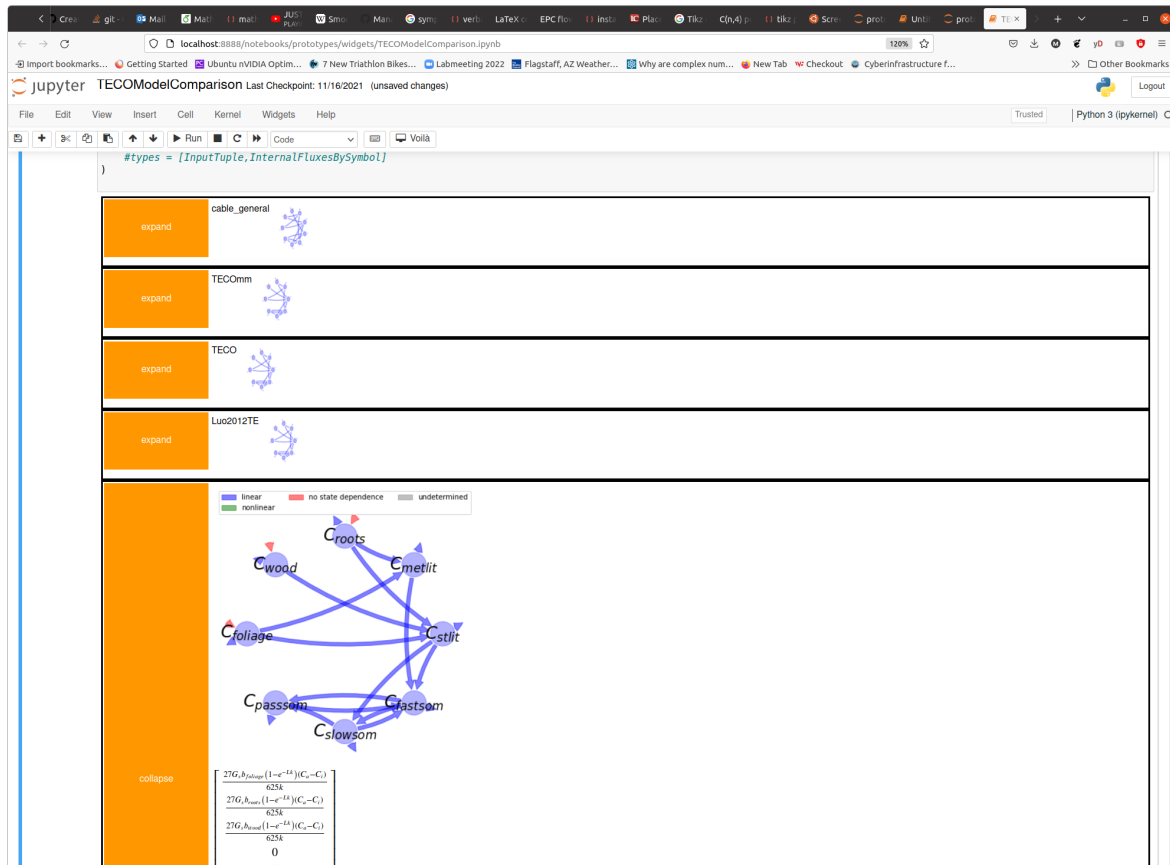
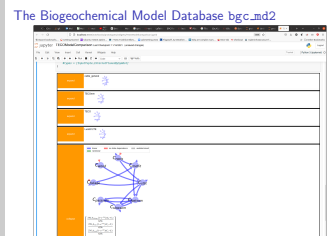


The Biogeochemical Model Database bgc_md2



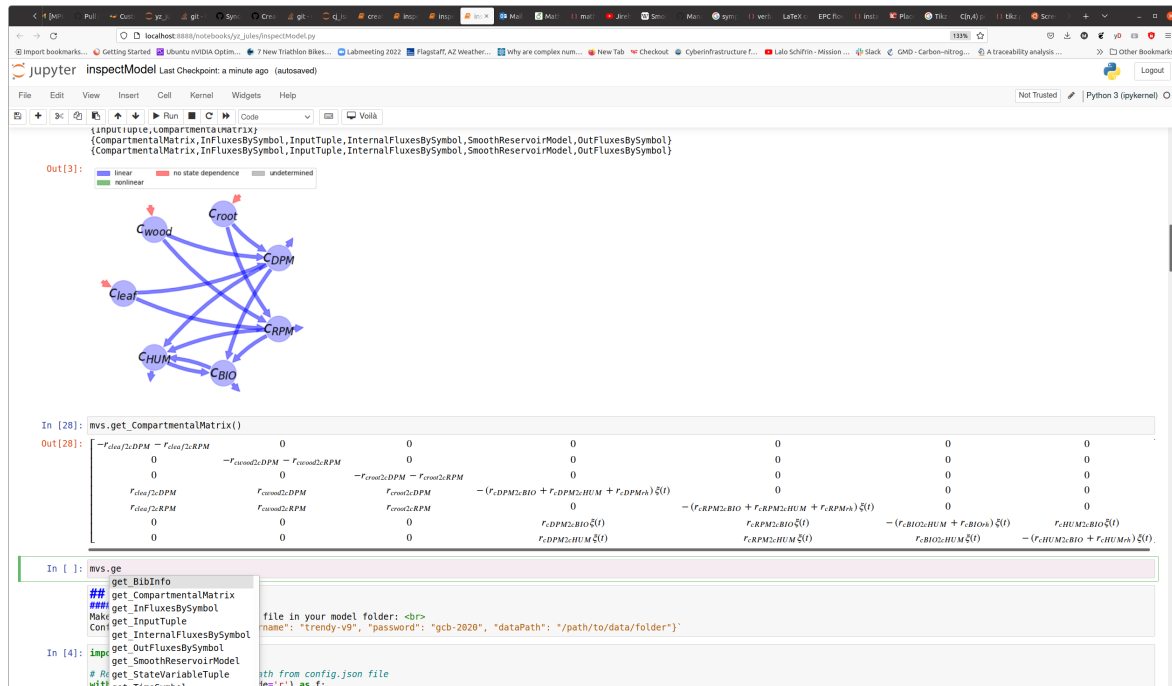
The Biogeochemical Model Database

└ The Biogeochemical Model Database bgc_md2



1. bgc_md2 is an open source python package available on GitHub, developed at the Max-Planck-Institut for BioGeoChemistry in Jena and more recently in Yiqi Luo's Ecolab at NAU in Flagstaff
2. A set of libraries that can be used in other python scripts or interactively (jupyter or IPython) The picture shows a jupyter widget showing a table of models. The orange buttons can be clicked to expand or collapse a more detailed view of the particular model.
3. > 30 published vegetation, soil or ecosystems models in a format that for symbolic and numeric computations
4. A set of special datatypes that describe components of the models and functions that operate on these datatypes
5. A userinterface that uses a graph library to compute what is computable and can be used for comparisons.

Analysis with symbolic tools ...

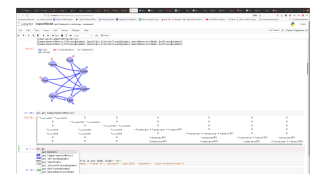


Navigation icons: back, forward, search, etc.

The Biogeochemical Model Database

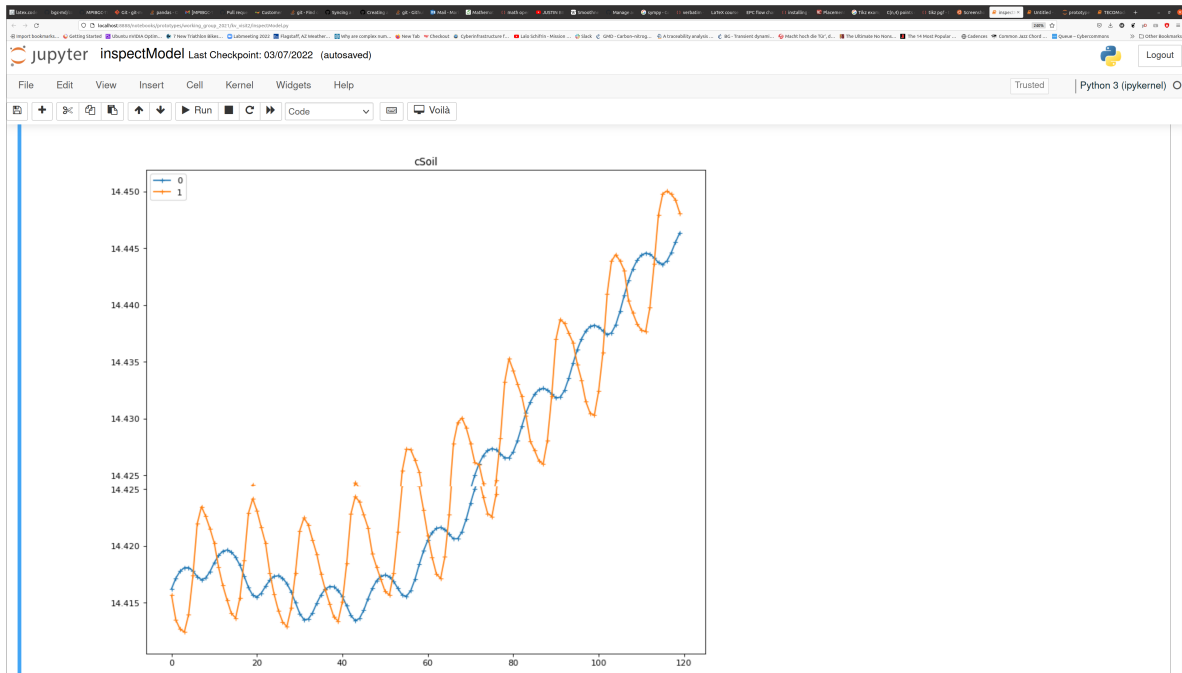
└ Analysis with symbolic tools ...

Analysis with symbolic tools ...



1. the structure (graph both in the mathematical and visual sense) can be derived from the symbolic description
2. other properties are flux equations the compartmental matrix

...or numerically



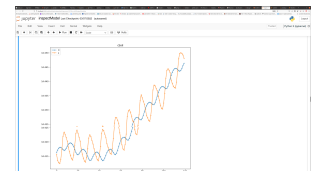
Navigation icons: back, forward, search, and other controls.

2022-03-21

The Biogeochemical Model Database

└ ...or numerically

...or numerically



1. the symbolic model description can be parameterized and transformed into a numeric model
2. The picture shows the data assimilation result for the above model using trendy data.

Diagnostic Variables implemented once, available for all models

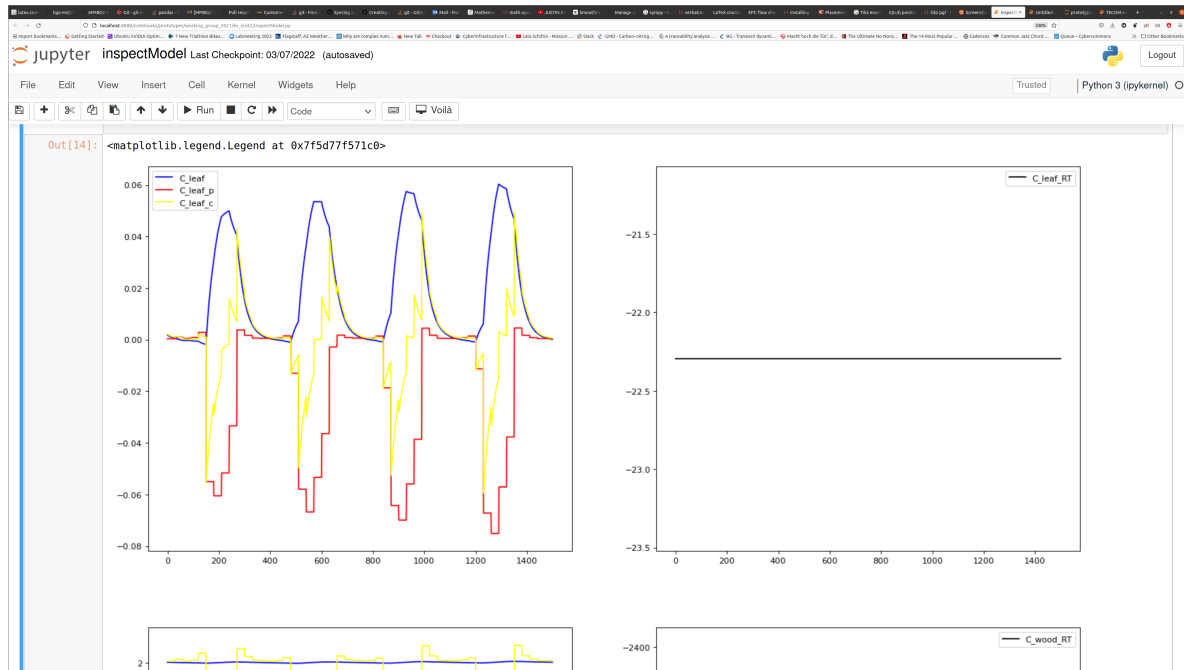


Figure: pool content + Traceability Analysis: carbon storage potential , carbon storage capacity and residence time

Navigation icons: back, forward, search, and other controls.

The Biogeochemical Model Database

2022-03-21

└ Diagnostic Variables implemented once, available for all models

Diagnostic Variables implemented once, available for all models

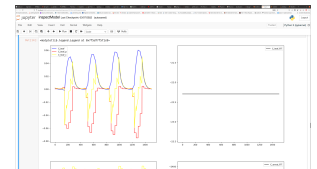


Figure: pool content + Traceability Analysis: carbon storage potential , carbon storage capacity and residence time

1. Diagnostic Variables can be computed for any model
2. The picture shows the Leaf pool

Userinterface using computability graphs

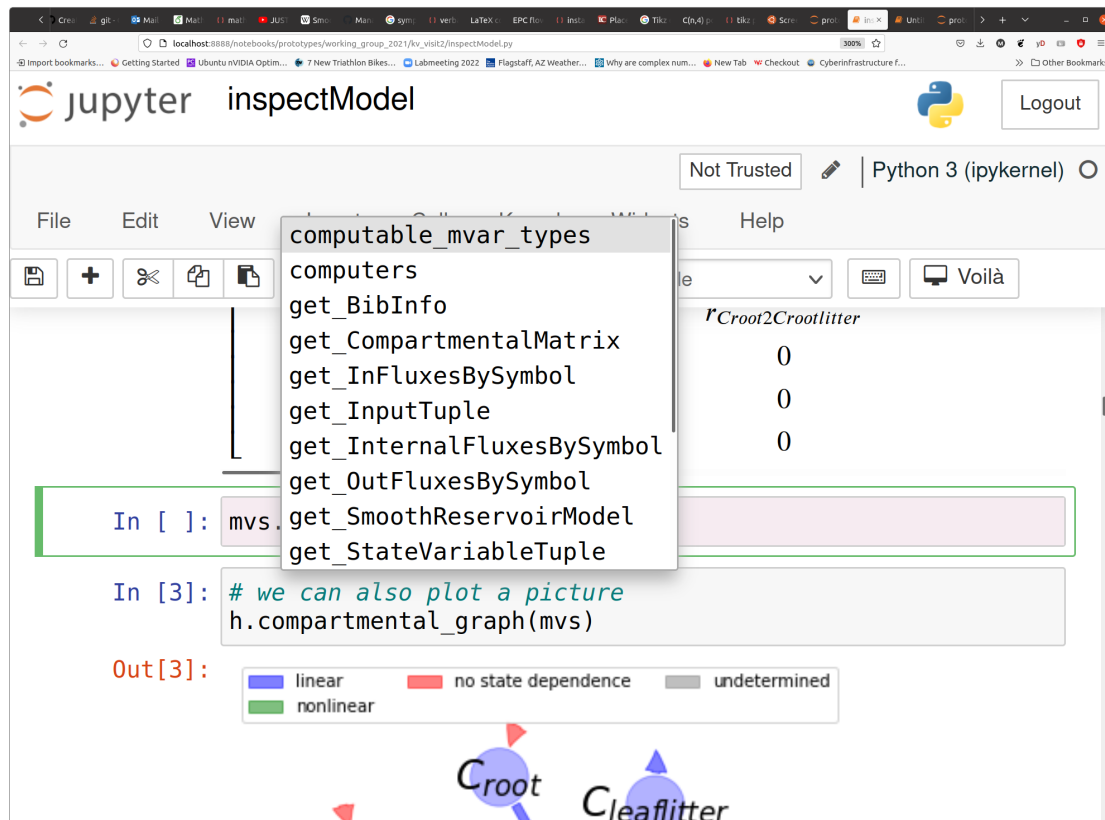


Figure: Suggested methods automatically created by a graph library

The Biogeochemical Model Database

Userinterface using computability graphs

1. The figure shows IPythons options for callable methods on a variable called `mvs`
2. This is standard for methods of a python object. In this case the methods are automatically created and added by a graphalgorithm that computes which variables can be computed form the set of provided model properties.
3. This has far reaching consequences. Models can be compared with respect to all variables in the "convex hull under computability", with respect to all *computable variables* not just the ones provided in the data base record.
4. As an example the matrix formulation of models A and B can be compared even if neither A nor B defines the matrix as long as it can be computed from other variable (in this case the internal and outfluxes and an ordering of the statevariables)
5. An obvious consequence is that the information about a model can be provided in different ways. There is no need to force the user into a rigid record format. Another consequence is that records do not have to be complete. The framework accepts all information about a model and (computes what it can do with it).
6. The set of computable properties can be used to query the database (e.g. which models have a vegetation part) ...

Finding what's missing

given a set of
functions:

$a(i)$, $b(c,d)$, $b(e,f)$,
 $c(b)$, $d(b)$, $d(g,h)$,
 $e(b)$, $f(b)$ and the
target variable **B**

e.g.

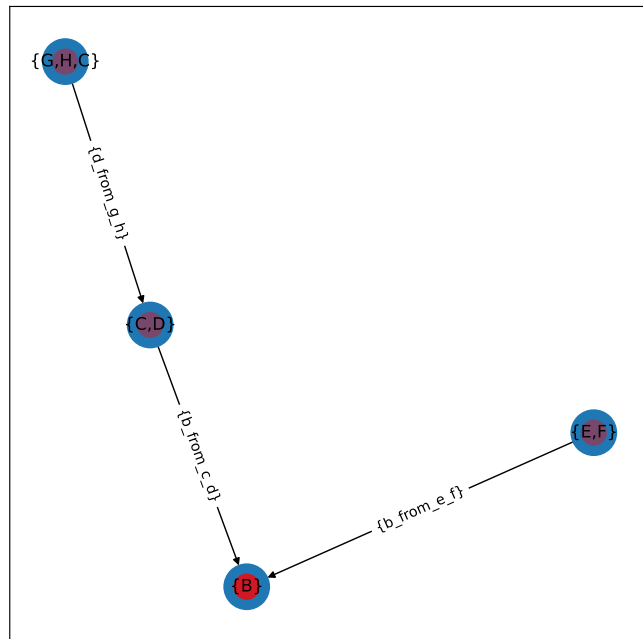
CompartmentalMatrix,

The algorithm

computes all

possible

combinations and
paths from which **B**
can be computed.



Navigation icons: back, forward, search, etc.

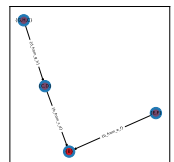
2022-03-21

The Biogeochemical Model Database

└ Finding what's missing

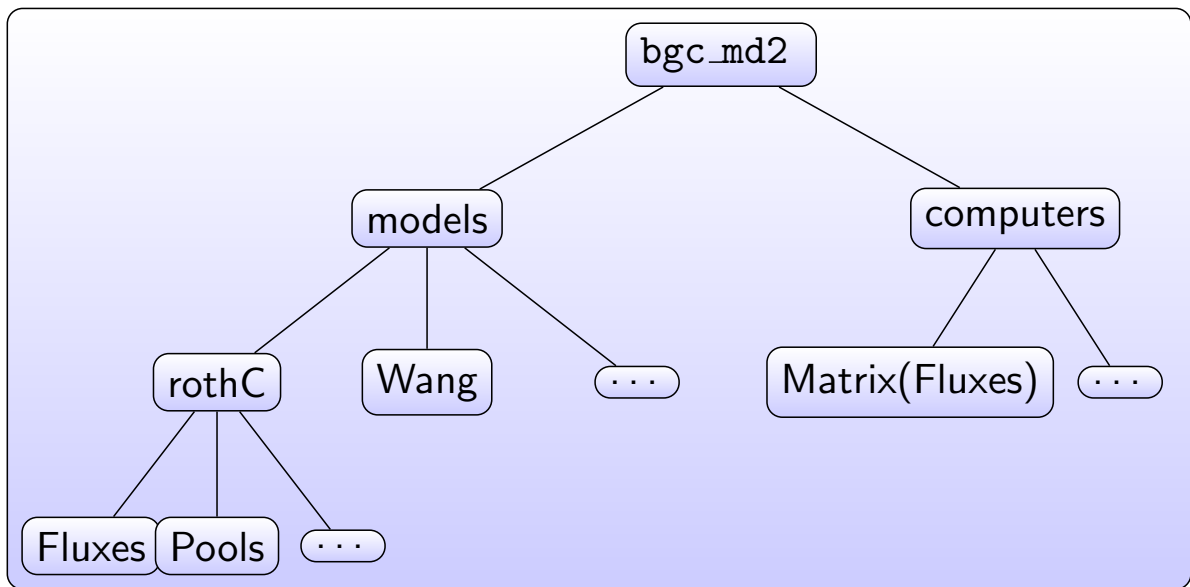
Finding what's missing

given a set of
functions:
 $a(i)$, $b(c,d)$, $b(e,f)$,
 $c(b)$, $d(b)$, $d(g,h)$,
 $e(b)$, $f(b)$ and the
target variable **B**
e.g.
CompartmentalMatrix.
The algorithm
computes all
possible
combinations and
paths from which **B**
can be computed.



1. It is not only possible to compute what can be computed from a given model description, It is also possible to do the opposite and ask the algorithm what missing information has to be provided to compute a target property of a model.
2. The algorithm finds a path (a combination of functions that can be applied recursively) and automatically computes intermediate results.

Internal Structure of bgc_md2



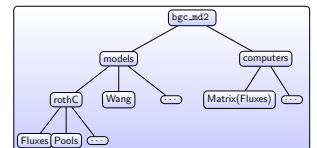
Navigation icons: back, forward, search, etc.

2022-03-21

The Biogeochemical Model Database

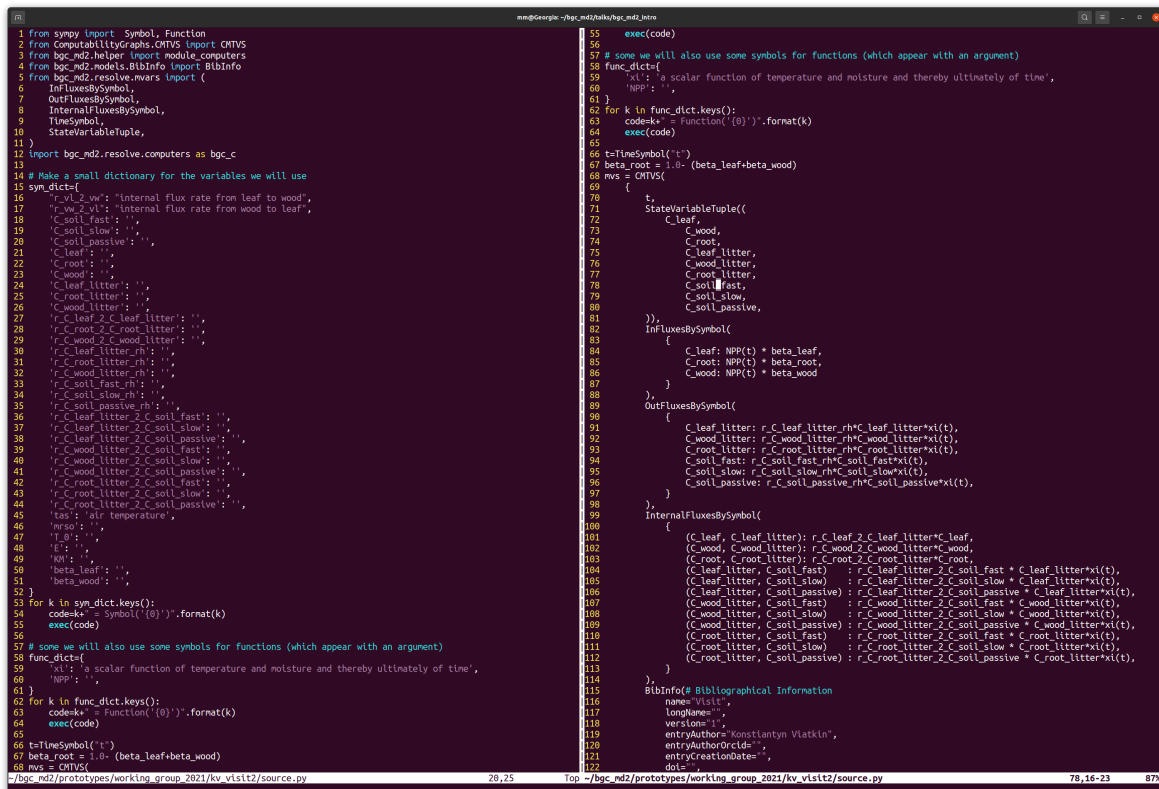
└ Internal Structure of bgc_md2

Internal Structure of bgc_md2



1. `bgc_md2` is not just a collection of models, described as sets of variable of special type like fluxes or matrice, but also a collection of functions whose arguments and return values have these types. These functions are here called `computers` and use python type annotations. The computability graph used in the user interface and queries is derived from the annotations of a set of functions. The set of properties (defined by the types) is growing as well as the functions connecting them.

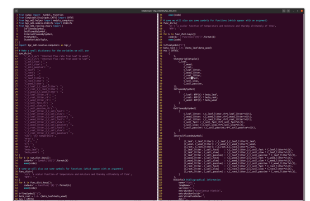
Database records are python modules

A screenshot of a Python script in a code editor. The script is divided into two main sections. The left section (lines 1-56) imports various modules like sympy, ComputerAlgebra, and CHNS, and defines a dictionary of model variables. The right section (lines 57-122) defines functions for calculating fluxes and internal fluxes, using the variables from the dictionary. The script uses a mix of standard Python syntax and sympy-specific functions for symbolic computation.

The Biogeochemical Model Database

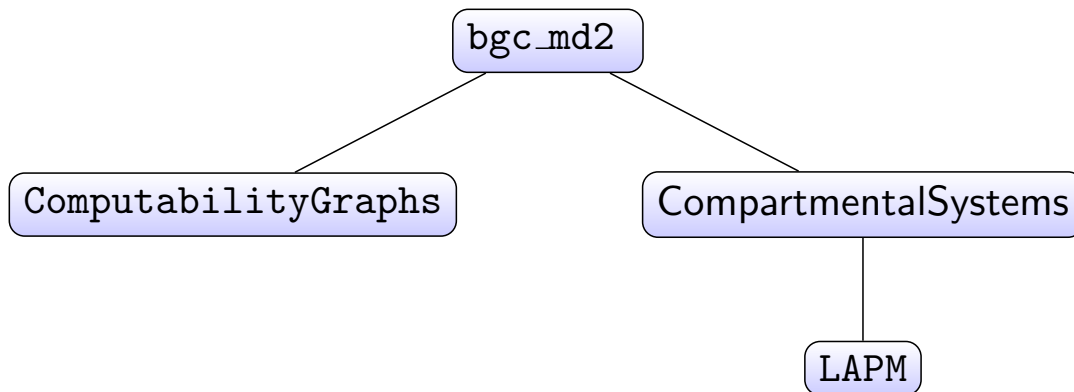
— Database records are python modules

Database records are python modules



1. The picture shows the screen shot of the source code of the above model using sympy and some datatypes provided by bgc_md2 .
2. The entries of the database dont even have to be complete models. They are implemented in normal python and have in common that they define a set of model properties and a set of functions to connect them. There is no special format necessary. The creation of the symbolic formulation can be automated bye all means available in python. Extra information can but does not have to be provided.

Relation to other Python Packages



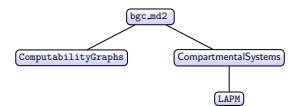
Navigation icons: back, forward, search, and other presentation controls.

2022-03-21

The Biogeochemical Model Database

└ Relation to other Python Packages

Relation to other Python Packages



1. The graph computation is outsourced into our package `ComputabilityGraphs`
2. Many of the advanced diagnostic variables (age and transittime distributions) are computed using our other packages `LAPM` and `CompartmentalSystems` for which `bgc_md2` acts as interface.

Applications

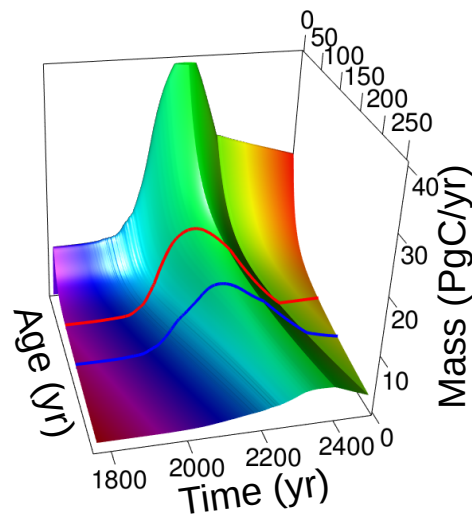


Figure: age distribution of a pool as function of time

Metzler, H., Müller, M., and Sierra, C. (2018). Transit-time and age distributions for nonlinear time-dependent compartmental systems. *Proceedings of the National Academy of Sciences*, 115:201705296.