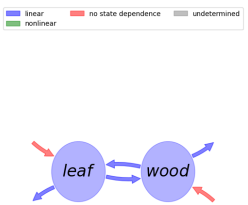
Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

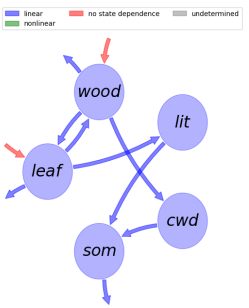
How to find the right model for a job?

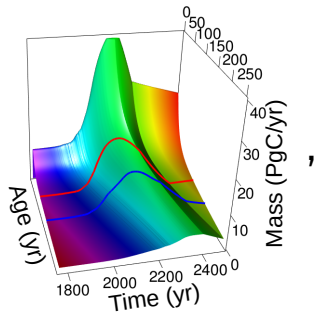
- We need Collections
  - of **models** e.g.: Arora2005GCB-1 ,CARDAMOM ,. . . , Zelenev2000MicrobialEcology
  - of **properties** and diagnostics to **compare** them by
    - ★ symbolic e.g. mass balance, CompartmentalMatrix, InFluxes . . .

$$\frac{d}{dt}\begin{bmatrix} leaf \\ wood \end{bmatrix} = \begin{bmatrix} I_{leaf}(t) \\ I_{wood} \end{bmatrix} + \begin{bmatrix} -k_{leaf2wood} - k_{leafo}(t) & k_{wood2leaf} \\ k_{leaf2wood} & -k_{wood2leaf} - k_{woodo} \end{bmatrix} \begin{bmatrix} leaf \\ wood \end{bmatrix}$$

    - ★ graphic, numeric e.g. carbon flux diagrams, transit times, age distributions . . .





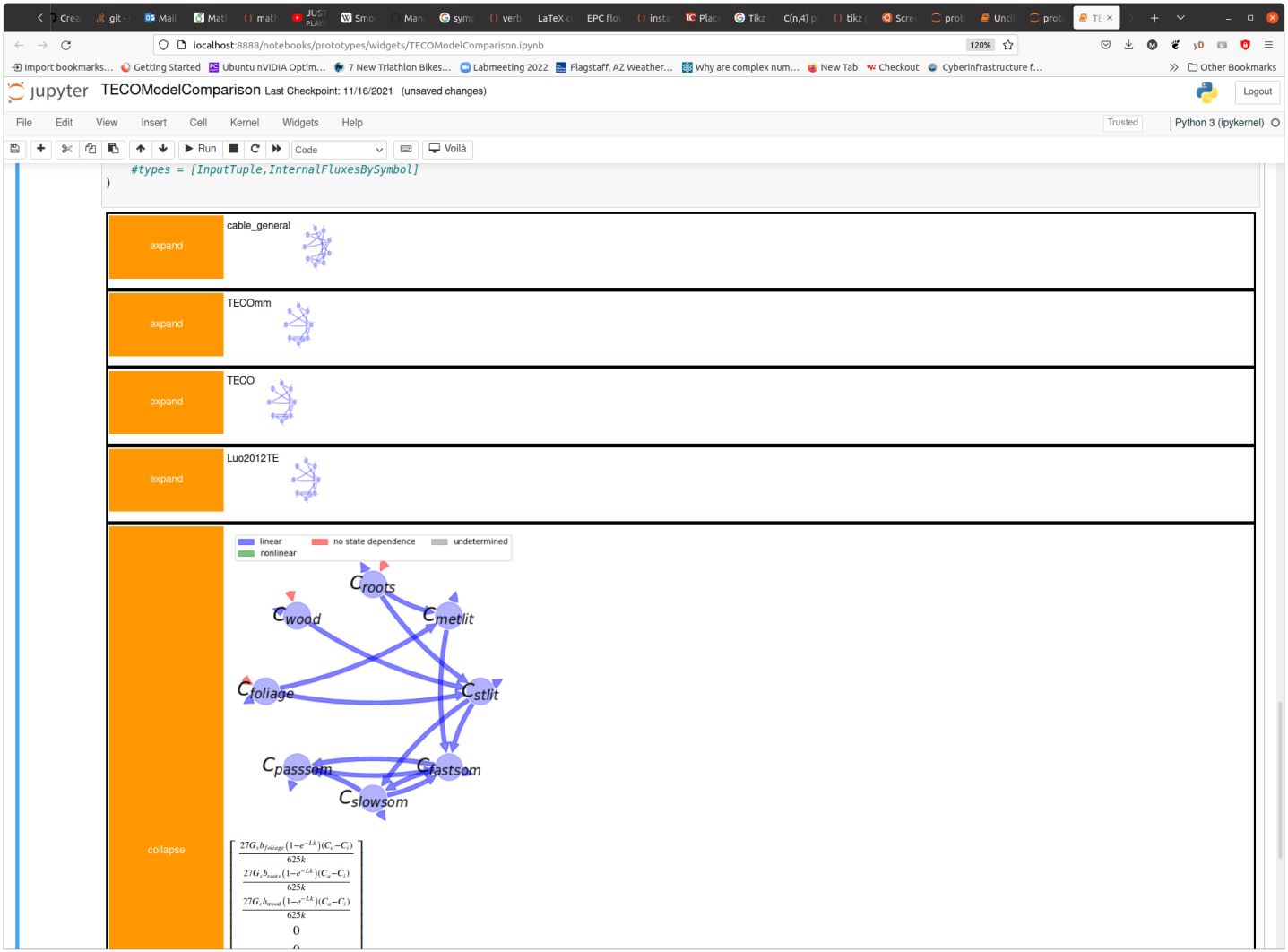


,

- **HOW** do we implement, inspect, test and extend both collections?
- **HOW** can you steal from it?

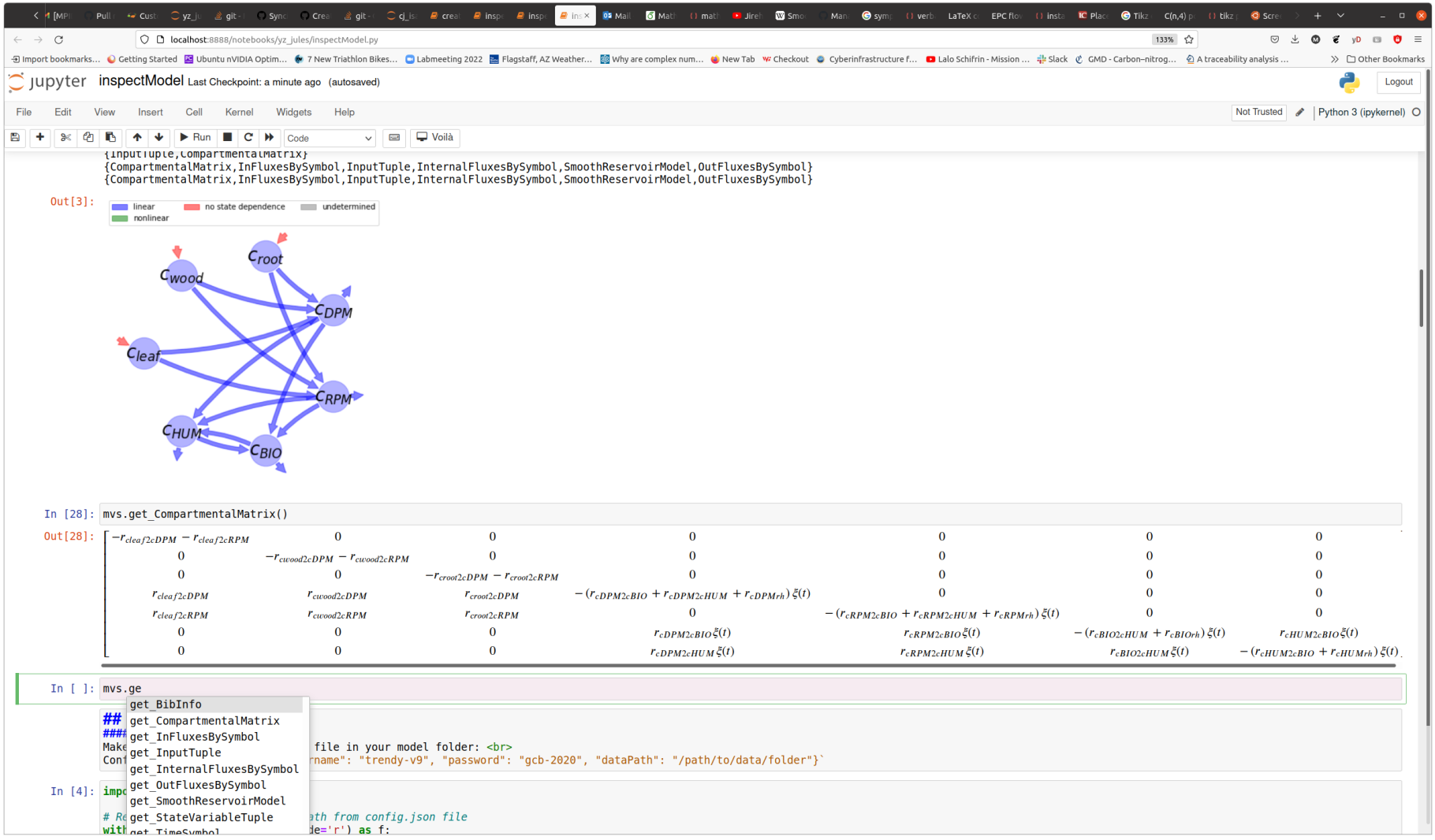
Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Example widget for query result



Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Analysis with symbolic tools . . .

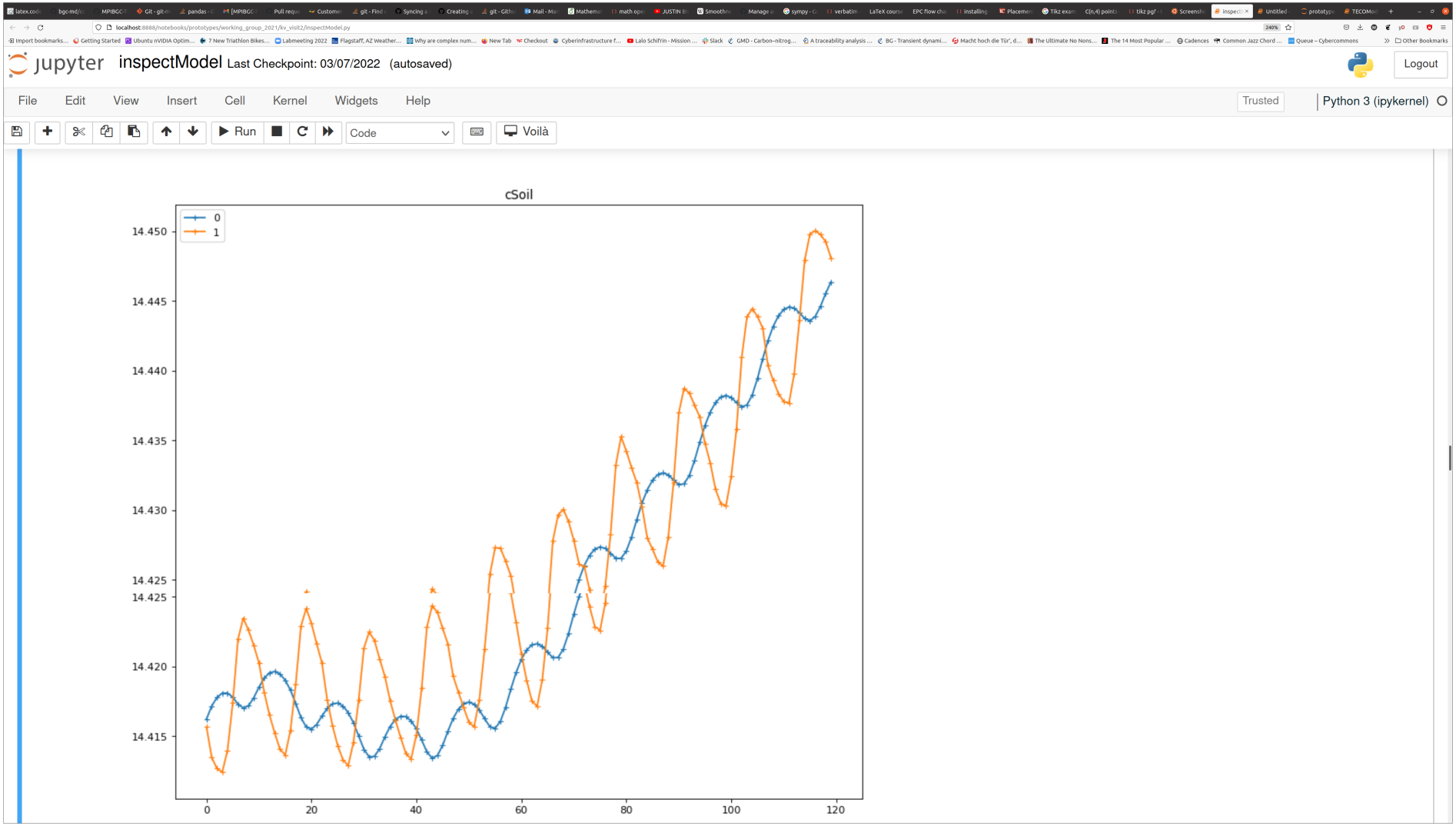


Presentation | Implementation | Screen Shots | Links

Reproducible Carbon Cycle Models
Biogeochemical Model Database bgc_md2

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

. . . or numerically



Reproducible Carbon Cycle Models
Biogeochemical Model Database bgc_md2

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Database records are python modules

mm@Georgia: ~/bgc_md2/talks/bgc_md2_intro

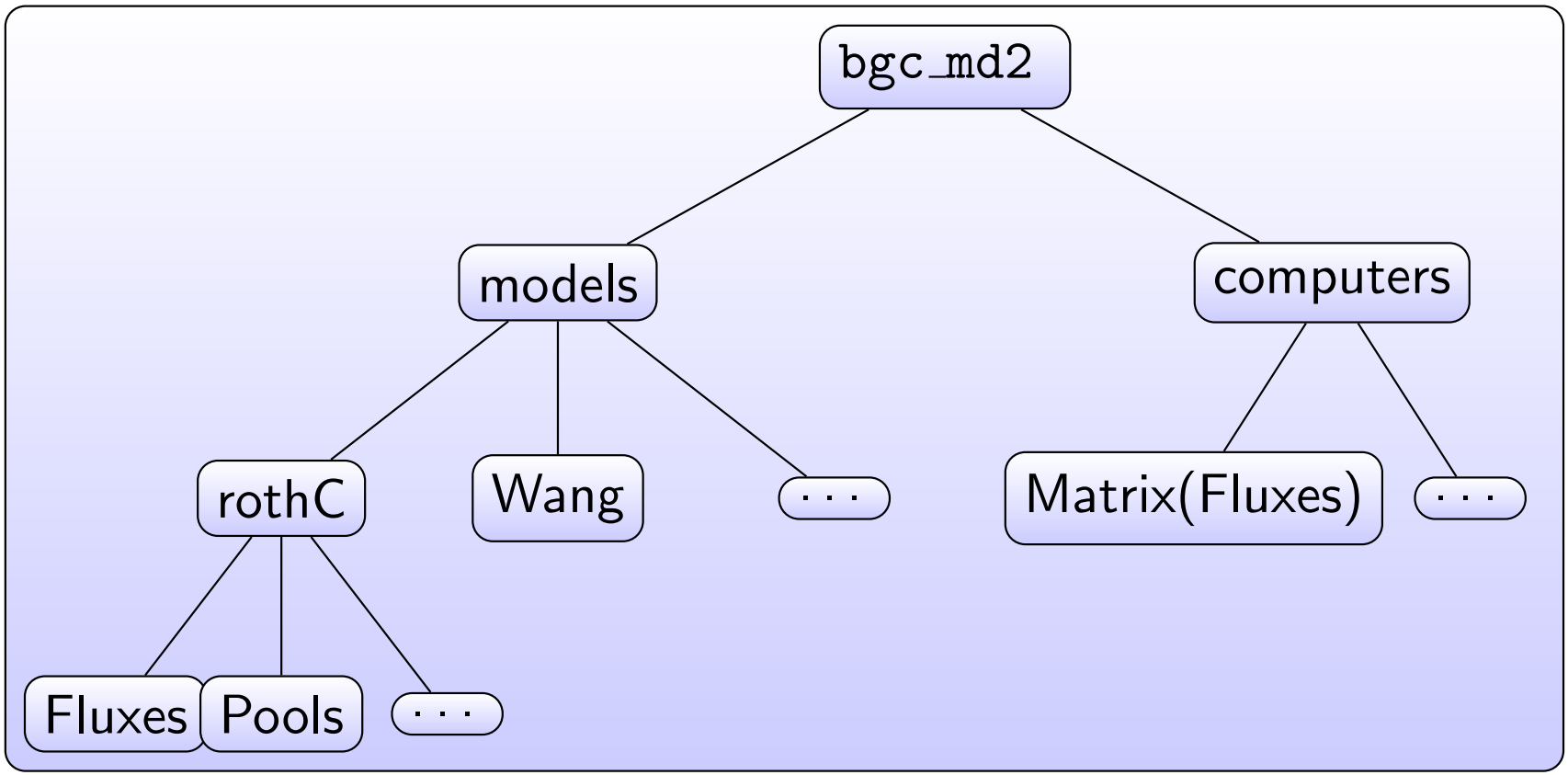
```python
from sympy import  Symbol, Function
from ComputabilityGraphs.CMTVS import CMTVS
from bgc_md2.helper import module_computers
from bgc_md2.models.BibInfo import BibInfo
from bgc_md2.resolve.mvars import (
    InFluxesBySymbol,
    OutFluxesBySymbol,
    InternalFluxesBySymbol,
    TimeSymbol,
    StateVariableTuple,
)
import bgc_md2.resolve.computers as bgc_c

# Make a small dictionary for the variables we will use
sym_dict={
    "r_vl_2_vw": "internal flux rate from leaf to wood",
    "r_vw_2_vl": "internal flux rate from wood to leaf",
    'C_soil_fast': '',
    'C_soil_slow': '',
    'C_soil_passive': '',
    'C_leaf': '',
    'C_root': '',
    'C_wood': '',
    'C_leaf_litter': '',
    'C_root_litter': '',
    'C_wood_litter': '',
    'r_C_leaf_2_C_leaf_litter': '',
    'r_C_root_2_C_root_litter': '',
    'r_C_wood_2_C_wood_litter': '',
    'r_C_leaf_litter_rh': '',
    'r_C_root_litter_rh': '',
    'r_C_wood_litter_rh': '',
    'r_C_soil_fast_rh': '',
    'r_C_soil_slow_rh': '',
    'r_C_soil_passive_rh': '',
    'r_C_leaf_litter_2_C_soil_fast': '',
    'r_C_leaf_litter_2_C_soil_slow': '',
    'r_C_leaf_litter_2_C_soil_passive': '',
    'r_C_wood_litter_2_C_soil_fast': '',
    'r_C_wood_litter_2_C_soil_slow': '',
    'r_C_wood_litter_2_C_soil_passive': '',
    'r_C_root_litter_2_C_soil_fast': '',
    'r_C_root_litter_2_C_soil_slow': '',
    'r_C_root_litter_2_C_soil_passive': '',
    'tas': 'air temperature',
    'mrso': '',
    'T_0': '',
    'E': '',
    'KM': '',
    'beta_leaf': '',
    'beta_wood': '',
}
for k in sym_dict.keys():
    code=k+" = Symbol('{0}')".format(k)
    exec(code)

# some we will also use some symbols for functions (which appear with an argument)
func_dict={
    'xi': 'a scalar function of temperature and moisture and thereby ultimately of time',
    'NPP': '',
}
for k in func_dict.keys():
    code=k+" = Function('{0}')".format(k)
    exec(code)

t=TimeSymbol("t")
beta_root = 1.0- (beta_leaf+beta_wood)
mvs = CMTVS(
    {
        t,
        StateVariableTuple((
            C_leaf,
                C_wood,
                C_root,
                C_leaf_litter,
                C_wood_litter,
                C_root_litter,
                C_soil_fast,
                C_soil_slow,
                C_soil_passive,
        )),
        InFluxesBySymbol(
            {
                C_leaf: NPP(t) * beta_leaf,
                C_root: NPP(t) * beta_root,
                C_wood: NPP(t) * beta_wood
            }
        ),
        OutFluxesBySymbol(
            {
                C_leaf_litter: r_C_leaf_litter_rh*C_leaf_litter*xi(t),
                C_wood_litter: r_C_wood_litter_rh*C_wood_litter*xi(t),
                C_root_litter: r_C_root_litter_rh*C_root_litter*xi(t),
                C_soil_fast: r_C_soil_fast_rh*C_soil_fast*xi(t),
                C_soil_slow: r_C_soil_slow_rh*C_soil_slow*xi(t),
                C_soil_passive: r_C_soil_passive_rh*C_soil_passive*xi(t),
            }
        ),
        InternalFluxesBySymbol(
            {
                (C_leaf, C_leaf_litter): r_C_leaf_2_C_leaf_litter*C_leaf,
                (C_wood, C_wood_litter): r_C_wood_2_C_wood_litter*C_wood,
                (C_root, C_root_litter): r_C_root_2_C_root_litter*C_root,
                (C_leaf_litter, C_soil_fast)    : r_C_leaf_litter_2_C_soil_fast * C_leaf_litter*xi(t),
                (C_leaf_litter, C_soil_slow)    : r_C_leaf_litter_2_C_soil_slow * C_leaf_litter*xi(t),
                (C_leaf_litter, C_soil_passive) : r_C_leaf_litter_2_C_soil_passive * C_leaf_litter*xi(t),
                (C_wood_litter, C_soil_fast)    : r_C_wood_litter_2_C_soil_fast * C_wood_litter*xi(t),
                (C_wood_litter, C_soil_slow)    : r_C_wood_litter_2_C_soil_slow * C_wood_litter*xi(t),
                (C_wood_litter, C_soil_passive) : r_C_wood_litter_2_C_soil_passive * C_wood_litter*xi(t),
                (C_root_litter, C_soil_fast)    : r_C_root_litter_2_C_soil_fast * C_root_litter*xi(t),
                (C_root_litter, C_soil_slow)    : r_C_root_litter_2_C_soil_slow * C_root_litter*xi(t),
                (C_root_litter, C_soil_passive) : r_C_root_litter_2_C_soil_passive * C_root_litter*xi(t),
            }
        ),
        BibInfo(# Bibliographical Information
            name="Vlsit",
            longName="",
            version="1",
            entryAuthor="Konstiantyn Vlatkin",
            entryAuthorOrcid="",
            entryCreationDate="",
            doi="",
```

~/bgc_md2/prototypes/working_group_2021/kv_visit2/source.py 20,25 Top

~/bgc_md2/prototypes/working_group_2021/kv_visit2/source.py 78,16-23 87%

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Internal Structure of bgc_md2



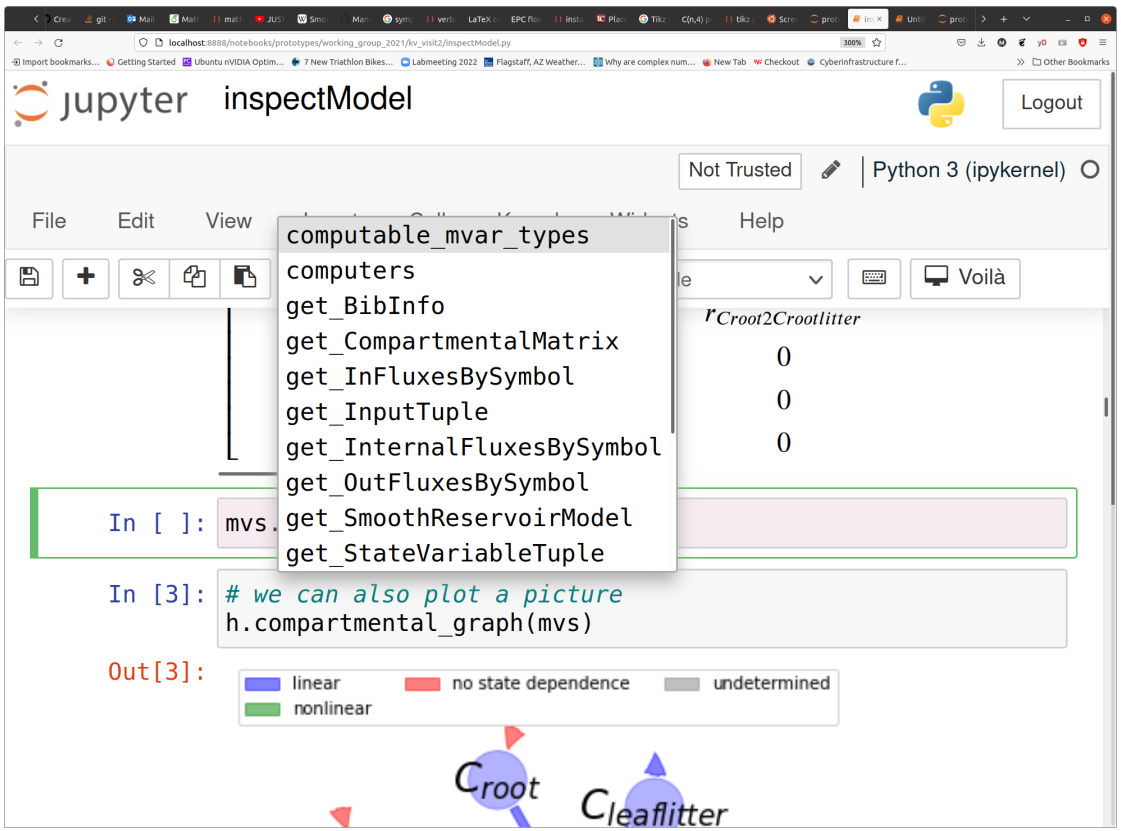
Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

The bgc_md library provides I:

1. Datatypes defining building blocks of models e.g. `CompartmentalMatrix`, `InternalFluxesBySymbol`, ...
2. Functions operating on those properties (forming the edges of the graph where the Datatypes are nodes)
3. A user interface based on graph algorithms to
   1. compute the set of computable properties (e.g. the comparable criteria for a set of models, database queries)
   2. actually compute the desired properties by recursively connecting several function applications.
   3. show what is missing to compute a desired property.

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Userinterface using computability graphs

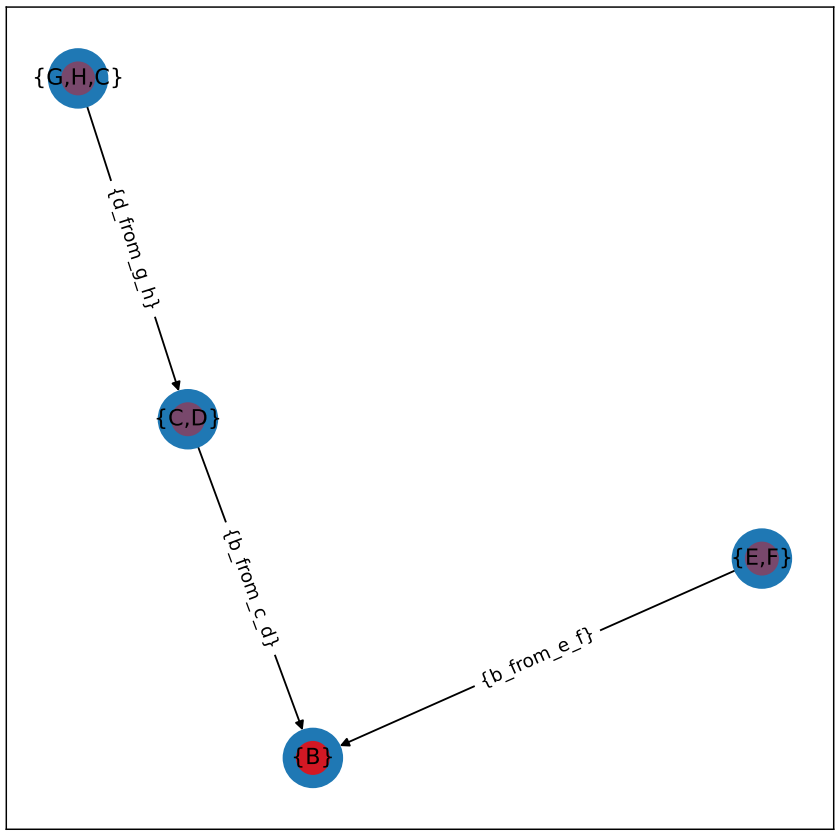


Suggested methods automatically created by a graph library

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Finding what's missing in the model description

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.



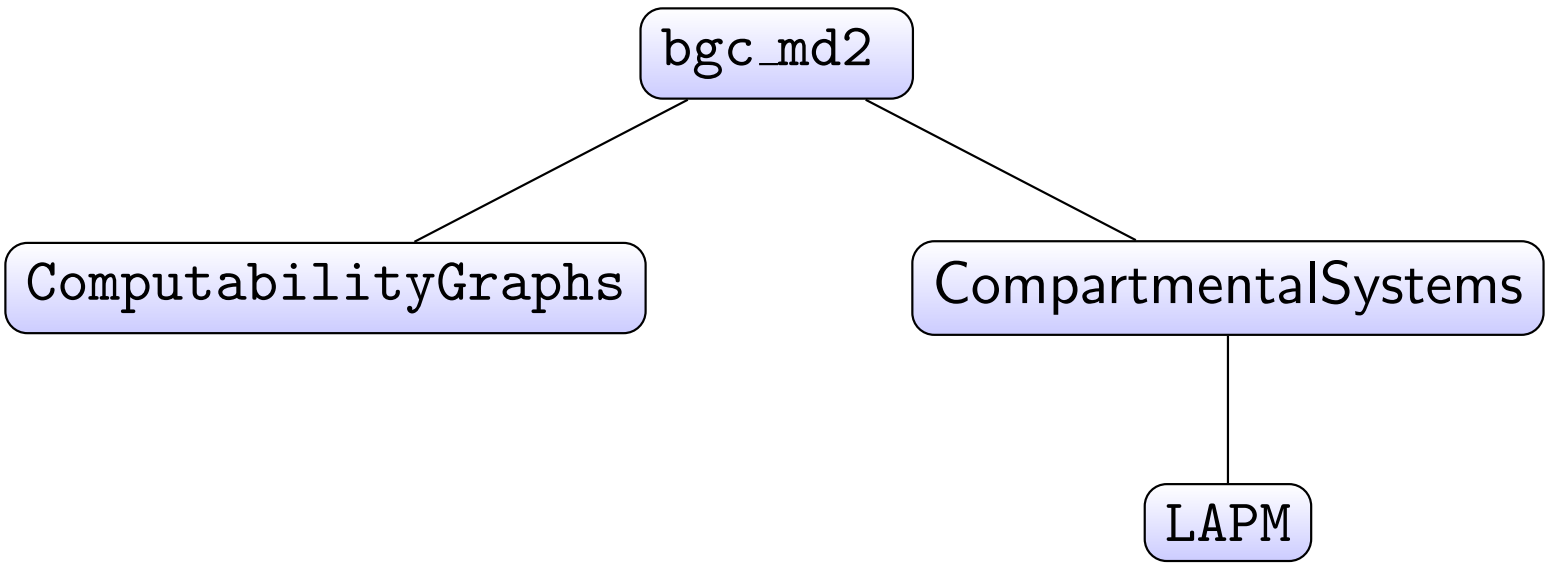
Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

The bgc_md library provides II:

1. 30+ vegetation, soil or ecosystem models for carbon and nitrogen cycling as reusable python modules using the building blocks in a flexible way.
2. An interface to *many algorithms* in `CompartmentalSystems` to compute diagnostic variables for *many models* in `bgc_md2`.

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Relation to other Python Packages



Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Example computation via CompartmentalSystems

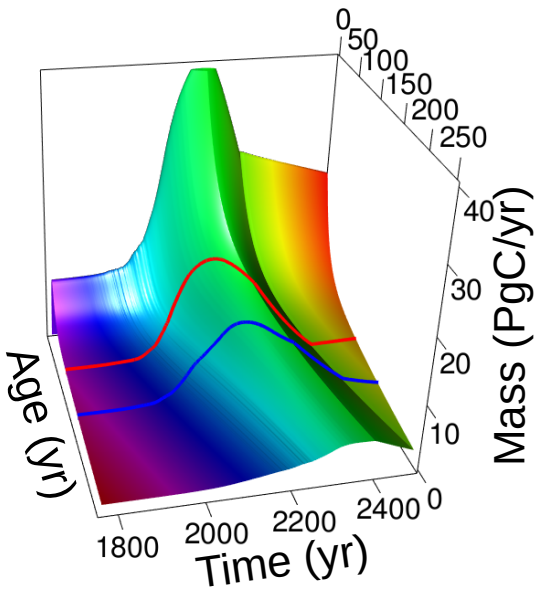


Figure: age distribuition of a pool as function of time

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.

Markus Müller, Holger Metzler, Verónika Ceballos Núñez, Kostiantyn Viatkin, Thomas Lotze, Jon Wells, Yu Zhou, Cuijuan Liao, Aneesh Chandel, Feng Tao, Yuanyuan Huang, Alison Bennett, Chenyu Bian, Lifen Jiang, Song Wang, Chengcheng Gang, Carlos Sierra, Yiqi Luo

Links

- The README of the package on github (wiht installation instructions): `https://github.com/MPIBGC-TEE/bgc_md2`
- `https://mybinder.org/v2/gh/MPIBGC-TEE/bgc_md2/binder` binder for testing some tutorials (jupyter notebooks) for the creation of new models or a model comparison No istallation neccessary.