bonsai_ipcc - a Python package for the calculation of national greenhouse gas inventories

Maik Budzinski [1], **Joao F. D. Rodrigues** [2], **and Mathieu Delpierre** [2]

1 Department of Sustainability and Planning, Aalborg University, Denmark. **2** 2.-0 LCA consultants, Aalborg, Denmark.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Pierre de Buyl

Reviewers:

- @GISWLH
- @PennyHow

Submitted: 17 November 2023
Published: unpublished

Summary

The aim of the bonsai_ipcc python package is to enable users to calculate national greenhouse gas (GHG) inventories based on the guidelines provided by the International Panel on Climate Change (IPCC) (Intergovernmental Panel on Climate Change, 2023). When implementing the equations and parameter data of these guidelines, the package follows the structure provided in the pdf documents (i.e., volumes and chapters). The package allows users to add their own data. Besides the calculation of default GHG inventories the software also implements tools for the calculation of error propagation, i.e., analytical error propagation and Monte Carlo simulation.

Statement of need

Gathering greenhouse gas (GHG) data is an important step when developing models and scenarios in many environmnetal sciences. The official guidelines for estimating national GHG inventories have been widely used in the modelling community e.g., to create environmentally extended input-output models (Merciai & Schmidt, 2018; Stadler et al., 2018) or datasets for life cycle assessment (Nemecek & Kaegi, 2007; Schmidt et al., 2021). The IPCC guidelines contain equations and default data that can be used to calculate country-based greenhouse gas inventories, taking into account different production and treatment activities.

However, calculating GHG inventories directly from the report is cumbersome and time consuming, requiring manual data extraction and visual inspection to identify the sequence of formulas that must be implemented.

To facilitate the compilation of GHG inventories, we developed an open-source Python package which stores the default data and implements the formulas of the IPCC report.

Structure of the package

The structure of bonsai_ipcc python package is illustrated in figure 1. The equations (in the following elementary equations) of a chapter are used to define the sequence (tier approach) to calculate the corresponding GHG inventory. Data for default parameter values of the guidelines is provided within the package. We use the Python package frictionless to describe and validate the provided data tables.

As a user, you choose the sequence and specify the dimensions (e.g., year, region) of the involved parameters. The result is a sequence of steps that store the involved parameter values and values that are calculated by elementary equations (represented by circles and rectangles, respectively in figure 1), as well as the involved uncertainty.

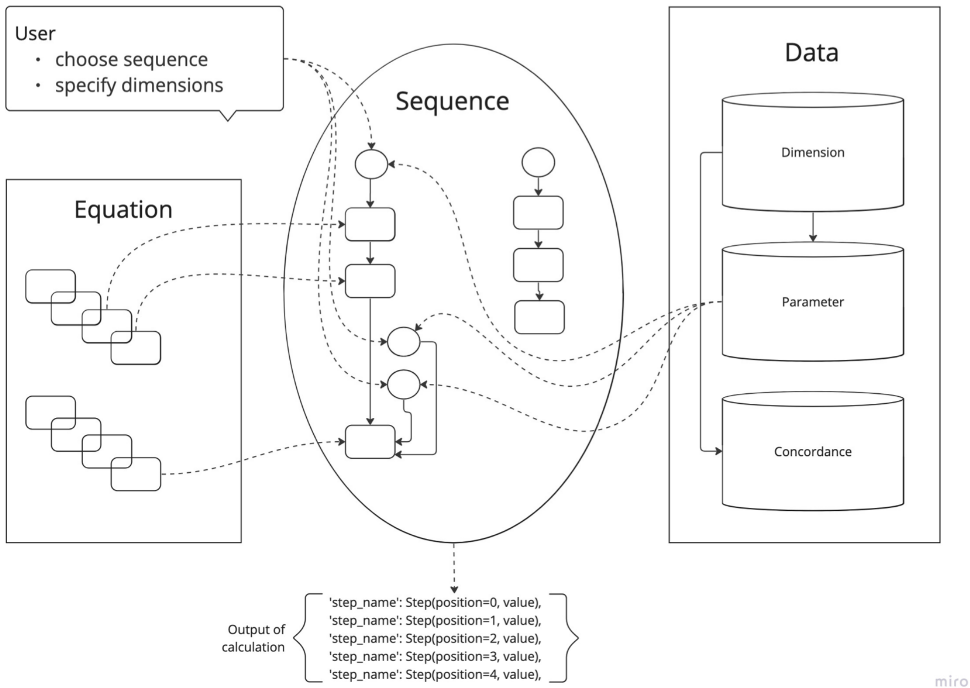

Figure 1: Structure of the bonsai_ipcc Python package

The package structure also follows the structure of the guidelines for estimating national GHG inventories. Each of the four core `<volume>`s (i.e., energy, agriculture, energy and waste) contains `<chapter>`s with elementary equations, which can be used to define the tier 1, 2 and 3 sequences calculating the inventories for GHG emissions (e.g., CO2, CH4 and N2O).

```
bonsai_ipcc.<volume>.<chapter>.sequence
bonsai_ipcc.<volume>.<chapter>.elementary
```

To distinguish between the different tiers 1, 2 and 3 when calculating the inventories for GHG emissions, the naming convention of the corresponding methods is as follows.

```
bonsai_ipcc.<volume>.<chapter>.sequence.tier<number>_<GHG>()
```

An example for such a sequence is given in the next section.

The organisation of tier methods (i.e., naming convention and definition of arguments) is a compromise between user and developer convenience. Keyword arguments of the tier methods follow the following principle.

```
tier<number>_<ghg>(year,region,<producttype>,<activitytype>,uncertainty)
```

Since the IPCC guidelines specify tier methods for each GHG separately, we decided to make this distinction in the name of the function instead of using an argument.

Core operation

The core feature of the bonsai_ipcc package is to determine GHG emissions for different tiers based on the provided data. The following code includes an example to calculate the CO2 emissions in chapter 5 (incineration and open burning of waste) of volume 5 (waste). Thereby, the emissions caused by the incineration of the plastic waste fraction of municipal waste are determined for continuous incineration by using the tier 1 approach.

The Journal of Open Source Software

Input:

```python
import bonsai_ipcc

my_ipcc = bonsai_ipcc.IPCC()
my_sequence = my_ipcc.waste.incineration.sequence.tier1_co2(
    year=2010, region="DE", wastetype= "msw_plastics",
    incintype= "continous", uncertainty="def")
my_sequence.__dict__
```

Output:

```
{'signature': {'year': 2010,
  'region': 'DE',
  'wastetype': 'msw_plastics',
  'incintype': 'continous',
  'uncertainty': 'def'},
'urb_population': Step(position=0, year=2010, unit='cap',
  value=62940432.0),
'MSW_gen_rate': Step(position=1, year=2010, unit='t/cap/yr',
  value=0.6),
'MSW_frac_to_incin': Step(position=2, year=2010, unit='kg/kg',
  value=0.37),
'MSW_type_frac': Step(position=3, year=2010, unit='kg/kg',
  value=0.104),
'SW_per_treat': Step(position=4, year=2010, unit='Gg/year',
  value=1453.1686940159996),
'incintype_frac': Step(position=5, year=2010, unit='kg/kg',
  value=1),
'SW_per_tech': Step(position=6, year=2010, unit='Gg/year',
  value=1453.1686940159996),
'dm': Step(position=7, year=None, unit='kg/kg', value=1.0),
'CF': Step(position=8, year=None, unit='kg/kg', value=0.75),
'FCF': Step(position=9, year=None, unit='kg/kg', value=1.0),
'OF': Step(position=10, year=None, unit='kg/kg', value=1.0),
'CO2_emissions': Step(position=11, year=2010, unit='Gg/year',
  value=3996.213908543999)}
```

The output is a sequence of steps stored in a dictionary. Each step includes the name of the parameter and its value and unit.

Data handling

The IPCC guidelines also provide default data for a large amount of parameters that are used in the elementary equations. This data is included in the python package. When including the data into the package, we follow the frictionless standards. These standards provide patterns to describe data, such as tables, files and datasets. The framework follows the five design principles - simplicity, extensibility, human-editable and machine-usable, reusable and applicable across different technologies. The parameter dimension and concordance tables are associated to the volume and chapter where these data is used.

```
bonsai_ipcc.<volume>.<chapter>.parameter
bonsai_ipcc.<volume>.<chapter>.dimension
bonsai_ipcc.<volume>.<chapter>.concordance
```

The data for parameters and dimensions is stored in tabular format as csv files. To query the values within the bonsai_ipcc package, we use pandas DataFrame.

Parameter tables are accessible as pandas DataFrames.

Input:

```
my_ipcc.waste.incineration.parameter.cf.head(5)
```

Output:

```
                              value   unit
region waste_type   property
World  msw_food     def        0.38  kg/kg
       msw_garden   def        0.49  kg/kg
       msw_paper    def        0.46  kg/kg
       msw_wood     def        0.50  kg/kg
       msw_textiles def        0.50  kg/kg
```

The dimension tables can be as well accessed as pandas DataFrames.

Input:

```
my_ipcc.waste.incineration.dimension.property
```

Output:

```
              description                  remarks
code
def               default                     mean
min               minimum          2.5th percentile
max               maximum         97.5th percentile
abs_max  absolute maximun  theoretical upper bound
abs_min  absolute minimum  theoretical lower bound
```

To automate the process of selecting the right parameter when building the tier sequences, the package uses the concept of concordance tables. Thereby, each attribute of a dimension, e.g. country `DE` in the dimension `region`, can be associated to other more aggregated attributes (e.g., `Western Europe`). This has the advantage that parameter values can be selected from other attributes in cases where the guidelines only provide data for more aggregated ones.

Input:

```
my_ipcc.waste.incineration.concordance.region.head()
```

Output:

```
                unregion geographicregion continent  world
country
AF         Southern Asia              NaN       ASI  World
AX       Northern Europe              NaN       EUR  World
AL       Southern Europe              NaN       EUR  World
DZ       Northern Africa              NaN       AFR  World
AS             Polynesia              NaN       AUS  World
```

When reading the values from a specific parameter table, the sequence algorithm first tries to find the dimension on the left hand side and proceeds stepwise to the right until a value is found. The same principle is used for other dimensions, including `year` and `<producttype>`.

Uncertainty

Two methods for uncertainty analysis are implemented in the `ipcc` package: analytical error propagation and Monte Carlo method. When running the sequence, the type of value in each step depends on the selected method for uncertainty calculation (`float` for `uncertainty="def"`, [ufloat] for `uncertainty="analytical"` and [numpy array] for `uncertainty="monte_carlo"`).

Analytical error propagation

Input:

```python
import bonsai_ipcc

my_ipcc = bonsai_ipcc.IPCC()
my_sequence = my_ipcc.waste.incineration.sequence.tier1_co2(
    year=2010, region="DE", wastetype= "msw_plastics",
    incintype= "continous", uncertainty="analytical")
my_sequence.__dict__
```

Output:

```
{'signature': {'year': 2010,
  'region': 'DE',
  'wastetype': 'msw_plastics',
  'incintype': 'continous',
  'uncertainty': 'analytical'},
'urb_population': Step(position=0, year=2010, unit='cap',
  value=62940432.0+/-642249.3061224493),
 'msw_gen_rate': Step(position=1, year=2010, unit='t/cap/yr',
   value=0.6+/-0.12244897959183673),
 'msw_frac_to_incin': Step(position=2, year=2010, unit='kg/kg',
   value=0.37+/-0.05663265306122448),
 'msw_type_frac': Step(position=3, year=2010, unit='kg/kg',
   value=0.104+/-0.015918367346938772),
 'sw_per_treat': Step(position=4, year=2010, unit='Gg/year',
   value=1453.1686940159996+/-432.5683475655835),
 'incintype_frac': Step(position=5, year=2010, unit='kg/kg',
   value=1.0+/-0),
 'sw_per_tech': Step(position=6, year=2010, unit='Gg/year',
   value=1453.1686940159996+/-432.5683475655835),
 'dm': Step(position=7, year=None, unit='kg/kg', value=1.0+/-0),
 'cf': Step(position=8, year=None, unit='kg/kg',
   value=0.76+/-0.04591836734693876),
 'fcf': Step(position=9, year=None, unit='kg/kg',
   value=0.975+/-0.012755102040816339),
 'of': Step(position=10, year=None, unit='kg/kg', value=1.0+/-0),
 'co2_emissions': Step(position=11, year=2010, unit='Gg/year',
   value=3948.259341641471+/-1200.3649954387145)}
```

Monte Carlo simulation

Input:

```python
import matplotlib.pyplot as plt
my_sequence = my_ipcc.waste.incineration.sequence.tier1_CO2(
    year=2010,region="Germany",wastetype="msw_plastics",
    incintype= "continous",uncertainty="monte_carlo")
plt.hist(sequence.CO2_emissions.value)
```

Output:

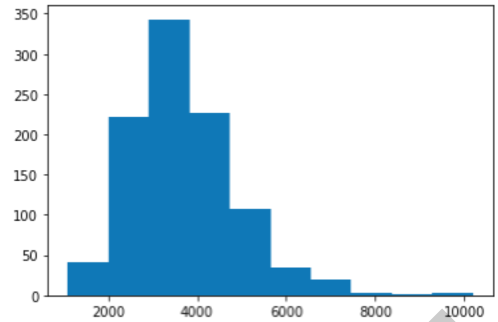

Figure 2: Monte Carlo result

Based on the provided uncertainty information for a parameter ("def", "min", "max", "abs_min", "abs_max"), the algorithm chooses the proper type of uncertainty distribution. The following distribution types are implemented, `normal`, `lognormal`, `truncated normal`, `uniform`, `truncated exponetial` and `beta` distribution. Truncated normal distributions are adjusted based on Rodrigues (2016) so that original mean and standard deviation are kept.

Conclusion

The transformation of the IPCC guidelines for calculation greenhouse gas inventories into the `bonsai_ipcc` Python package is an important step towards reproducability and automatization of national GHG inventory results. Furthermore, users of the package can use the results when developing models and scenarios in different scientific fields. Due to the magnitute of the IPCC guidelines, the implemenetation of its volumes into the Python package is an ongoing process. To this date one volume (waste) out of the four core volumes has been fully implemented. A second one (agriculture) is in progress. The implementation of a third one (industry) has been started. And a fourth (energy) is waiting to be initialized.

Acknowledgments

This project has received funding from the KR Foundation.

References

Intergovernmental Panel on Climate Change. (2023). *2019 Refinement to the 2006 IPCC Guidelines for National Greenhouse Gas Inventories*. https://www.ipcc-nggip.iges.or.jp/public/2019rf/index.html

Merciai, S., & Schmidt, J. (2018). Methodology for the construction of global multi-regional hybrid supply and use tables for the EXIOBASE v3 database. *Journal of Industrial Ecology*, *22*(3), 516–531. https://doi.org/10.1111/jiec.12713

Nemecek, T., & Kaegi, T. (2007). *Ecoinvent report no. 15 - life cycle inventories of agricultural production systems*. https://db.ecoinvent.org/reports/15_Agriculture.pdf

Rodrigues, J. D. F. (2016). Maximum-entropy prior uncertainty and correlation of statistical economic data. *Journal of Business & Economic Statistics*, *34*(3), 357–367. https://doi.org/10.1080/07350015.2015.1038545

Schmidt, J., Merciai, S., Munoz, I., Rosa, M. de, & Astudillo, M. (2021). *The big climate database v1 – methodology report*. https://denstoreklimadatabase.dk/sites/

klimadatabasen.dk/files/media/document/Methodology%20report%20-%20The%20Big%20Climate%20Database%20-%20Version%201.pdf

Stadler, K., Wood, R., Bulavskaya, T., Södersten, C.-J., Simas, M., Schmidt, S., Usubiaga, A., Acosta-Fernández, J., Kuenen, J., Bruckner, M., Giljum, S., Lutter, S., Merciai, S., Schmidt, J. H., Theurl, M. C., Plutzar, C., Kastner, T., Eisenmenger, N., Erb, K.-H., … Tukker, A. (2018). EXIOBASE 3: Developing a time series of detailed environmentally extended multi-regional input-output tables. *Journal of Industrial Ecology*, *22*(3), 502–515. https://doi.org/10.1111/jiec.12715