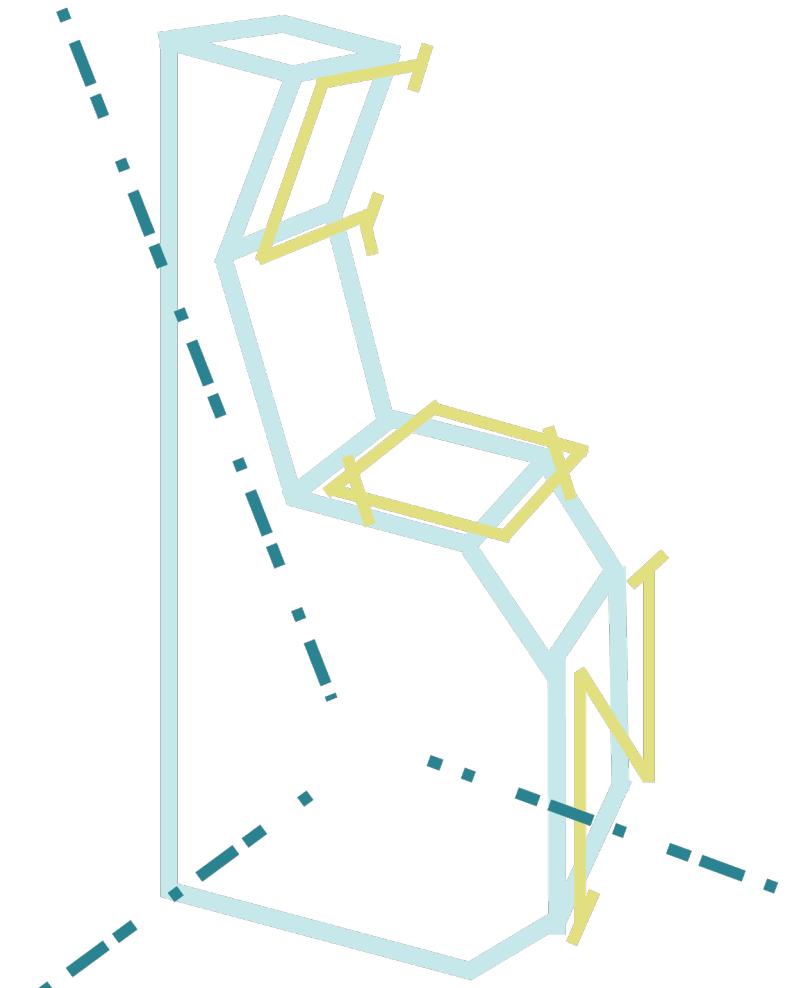


Brightway Documentation and Brightway in the Browser

Brightcon 2023, Luxembourg

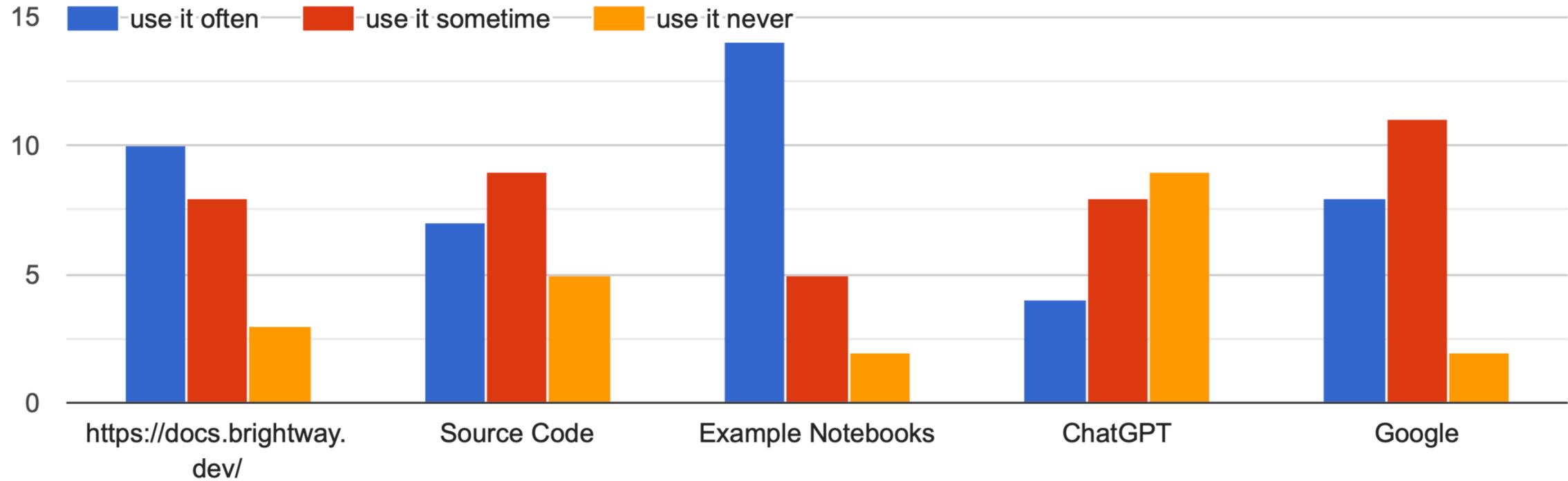
Michael Weinold

Doctoral Researcher at ETH Zurich



Why Documentation?

“If you got stuck with Brightway functionality, where do you usually go for help?”
Responses from ~20 participants in a Brightway workshop in February 2023.



What has changed?

Installation
Introduction
Example notebooks
Known issues
Development blog
Test coverage
CORE PACKAGES
Documentation
Data
Calc
Input-Output
Analyzer
Stats arrays
Speedups
Parameters
EXTENSIONS
Regional
Temporalis
Pedigree matrix
Activity browser



Advanced life cycle assessment framework

Introduction

Brightway2 is a open source framework for [life cycle assessment \(LCA\)](#). It doesn't try to replace software like SimaPro or OpenLCA, but rather to offer new possibilities to those who want to break the limits of conventional LCA. So it's not for everyone, and you will need to invest some time learning about Python to use Brightway2. Basically, this is the framework for you if your project lies at the intersection of LCA and your imagination.

Features

- More than 100 Monte Carlo iterations/second
- Use with [Jupyter notebooks](#)
- Downstream and upstream supply chain traversal
- Multiple database backends
- Graphs in Python ([Matplotlib](#)) or Javascript ([D3.js](#))

Principles

- **Open.** All code in Brightway2 is open source, and made available under the [BSD license](#).
- **Agile.** The simplicity of the data format and calculation engine make it easy to adapt the code or add new functionality.
- **Freedom.** A inventory dataset is a document with a few required fields, and can be adapted to your particular model or problem domain.
- **Modular.** Each set of functions is split into a package, with its own tests and documentation.

```
475      The building blocks in Brightway2 are LCI databases, LCIA methods, etc. However, we also need to keep track of which LCI databases and L
476
477      The base class for metadata is :ref:`serialized-dict`, which is basically a normal Python dictionary that can be easily saved or loaded
478
479      Brightway2 defines the following metadata stores:
480
481      * :ref:`databases`: LCI databases
482      * :ref:`methods`: LCIA methods (characterization factors)
483      * :ref:`normalizations`: LCIA normalization factors
484      * :ref:`weightings`: LCIA weighting factors
485
486      ta should be singletons
487
488      should be only one instance of each metadata store, to avoid having conflicting data (the `singleton pattern <http://en.wikipedia.org>`_).
489
490      e-block:: python
491
492      ass MyObjects(bw2data.serialization.SerializedDict):
493          file = "sweet-peppers.json"
494
495          objects = MyObjects()
496
497          metadata stores
498
499      ta stores are mostly useful when examining which objects are available:
500
501      e-block:: python
502
503      name in databases
504      print name
505      database name" in databases
506
507      ta stores are also used when deleting data objects:
508
509      e-block:: python
510
511      databases["some database to delete"]
512
513      y, and hopefully not surprisingly, metadata stores can be used to get the actual data object metadata:
514
515      e-block:: python
```

Internet Archive: Brightway Documentation in early 2022

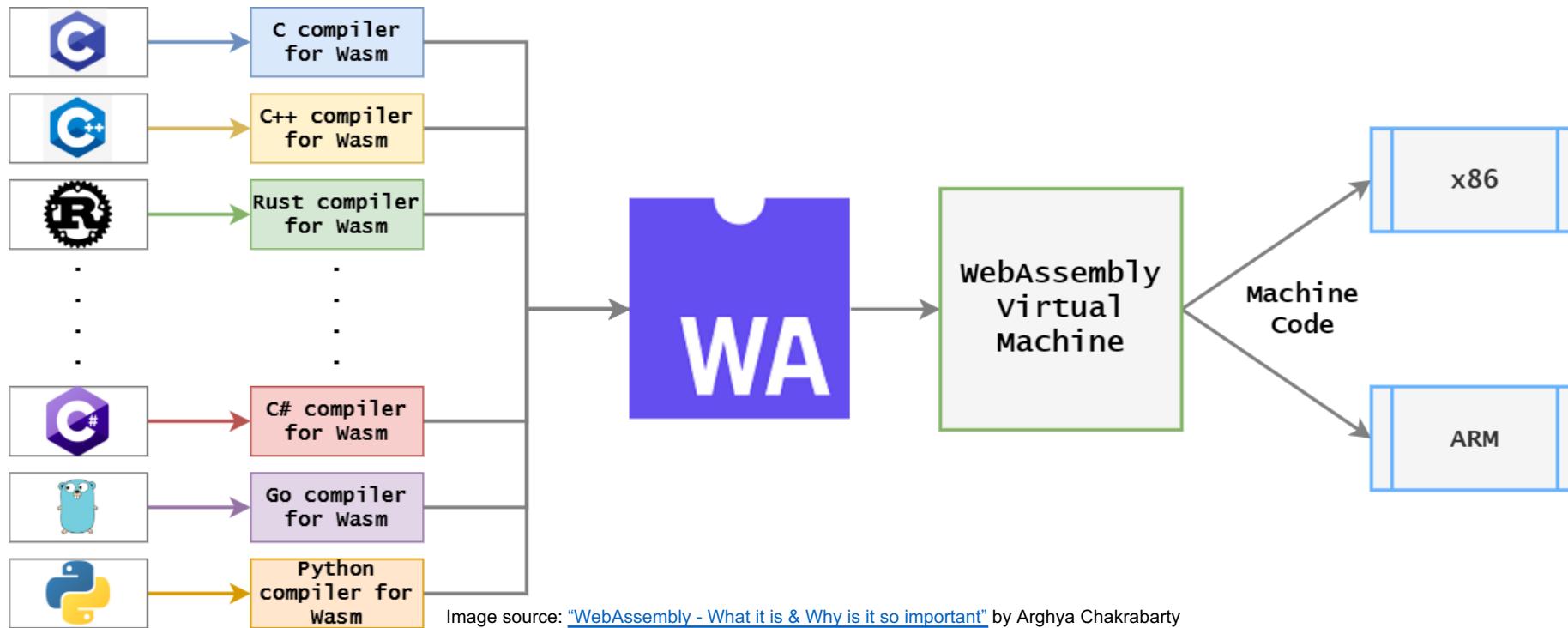
This is great, but.. .



Image by Wikimedia Commons user Stougaard under CC BY-SA 3.0

Web Assembly and Python in the Browser

WASM is a standard instruction format to enable high-performance applications on web pages.



Who uses WASM?

SciPy



 pandas



Brightway

Section Navigation

Lines, bars and markers

- Bar color demo
- [Bar Label Demo](#)
- Stacked bar chart
- Grouped bar chart with labels
- Horizontal bar chart
- Broken Barh
- CapStyle
- Plotting categorical variables
- Plotting the coherence of two signals
- Cross spectral density (CSD)
- [Curve with error band](#)
- Errorbar limit selection
- Errorbar subsampling
- EventCollection Demo
- Eventplot demo
- Filled polygon
- Fill Between and Alpha
- Filling the area between lines
- Fill Betweenx Demo
- Hatch-filled histograms
- Bar chart with gradients
- Hat graph
- Discrete distribution as horizontal bar chart
- JoinStyle
- Customizing dashed line styles
- Lines with a ticked patheffect

Curve with error band

This example illustrates how to draw an error band around a parametrized curve.

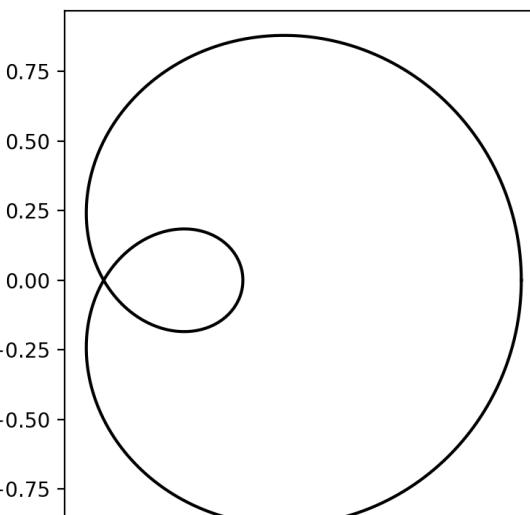
A parametrized curve $x(t), y(t)$ can directly be drawn using [plot](#).

```
import matplotlib.pyplot as plt
import numpy as np

from matplotlib.patches import PathPatch
from matplotlib.path import Path

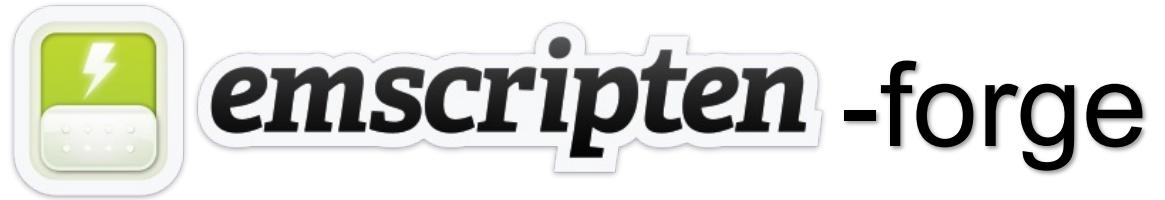
N = 400
t = np.linspace(0, 2 * np.pi, N)
r = 0.5 + np.cos(t)
x, y = r * np.cos(t), r * np.sin(t)

fig, ax = plt.subplots()
ax.plot(x, y, "k")
ax.set(aspect=1)
```



For completeness...

Brightway Live can use either the Pyodide distribution (monolithic) or the emscripten-forge distribution (similar to conda). Currently, Pyodide is more stable, so the website uses that.



We expect that future features like parquet support will run in WASM as well.



A screenshot of a browser window displaying a Python console interface. The title bar shows the URL `pyodide.org/en/stable/console`. The console output is as follows:

```
bike = 103
co2 = 201
>>> lca = bw2calc.LCA(
    demand={bike: 1},
    data_objs=[dp_static],
)
>>> lca.lci()
lca.lcia()
lca.score
66.5
>>>
```

The text "jQuery Terminal" is visible in the top right corner of the console area.

Thank you! Get in touch?

[@michaelweinold](#) (GitHub)

[@michaelweinold](#) (𝕏)

[in/michaelweinold](#) (LinkedIn)

[dev@weinold.ch](#) (E-Mail)

Please contribute to the Documentation...

<https://github.com/brightway-lca/brightway-documentation>

<https://github.com/brightway-lca/brightway-book>

<https://github.com/brightway-lca/brightway-live>