

Brightway 2.5

Features and release schedule

Chris Mutel, 20.10.2020

Photo source

Two major defects of Brightway2



**Poor support for
alternative backends**

- Default backend merges activities and products
- bw2io strongly linked to database schema in bw2data
- Need to support:
 - Other database schemas
 - Graph databases
 - Other open IE/LCA ontologies
- Main topic for Brightway version 3

Two major defects of Brightway2

- bw2io imports bw2data
- bw2calc imports bw2data
 - Mapping dict
 - Database dependency graph
- All Python-specific
 - Barrier to “web of models”
 - Main topic for Brightway version 2.5



Photo source

**No clean interfaces
between libraries**

Two major defects of Brightway2

- ~~brightway imports bw2data~~
Brightway version 3
- bw2calc imports bw2data
 - Mapping dict
 - Database dependency graph
- All Python-specific
 - Barrier to “web of models”
- Main topic for Brightway version 2.5



Photo source

**No clean interfaces
between libraries**

Removing mapping from bw2calc

mapping links objects (activities and locations) to unique integer ids

```
{  
    (“database name”, “code”): 1,  
    “DE”: 2  
}
```

Needed because we store processed data in *numerical* arrays

Removing mapping from bw2calc

mapping links objects (activities and locations) to unique integer ids

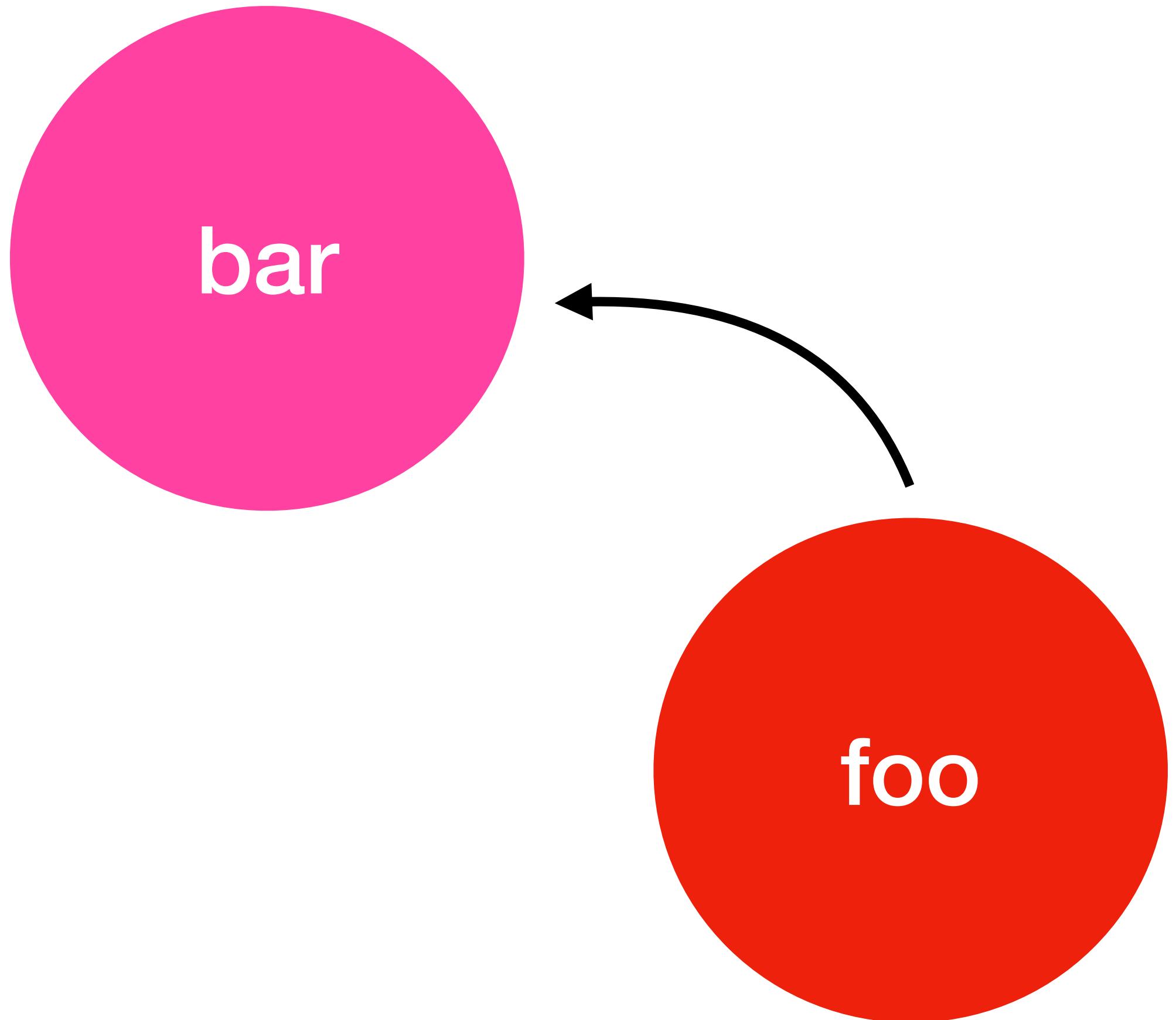
```
{  
    ("database name", "code"): 1,  
    "DE": 2  
}
```

Needed because we store processed data in *numerical* arrays

Strategy:

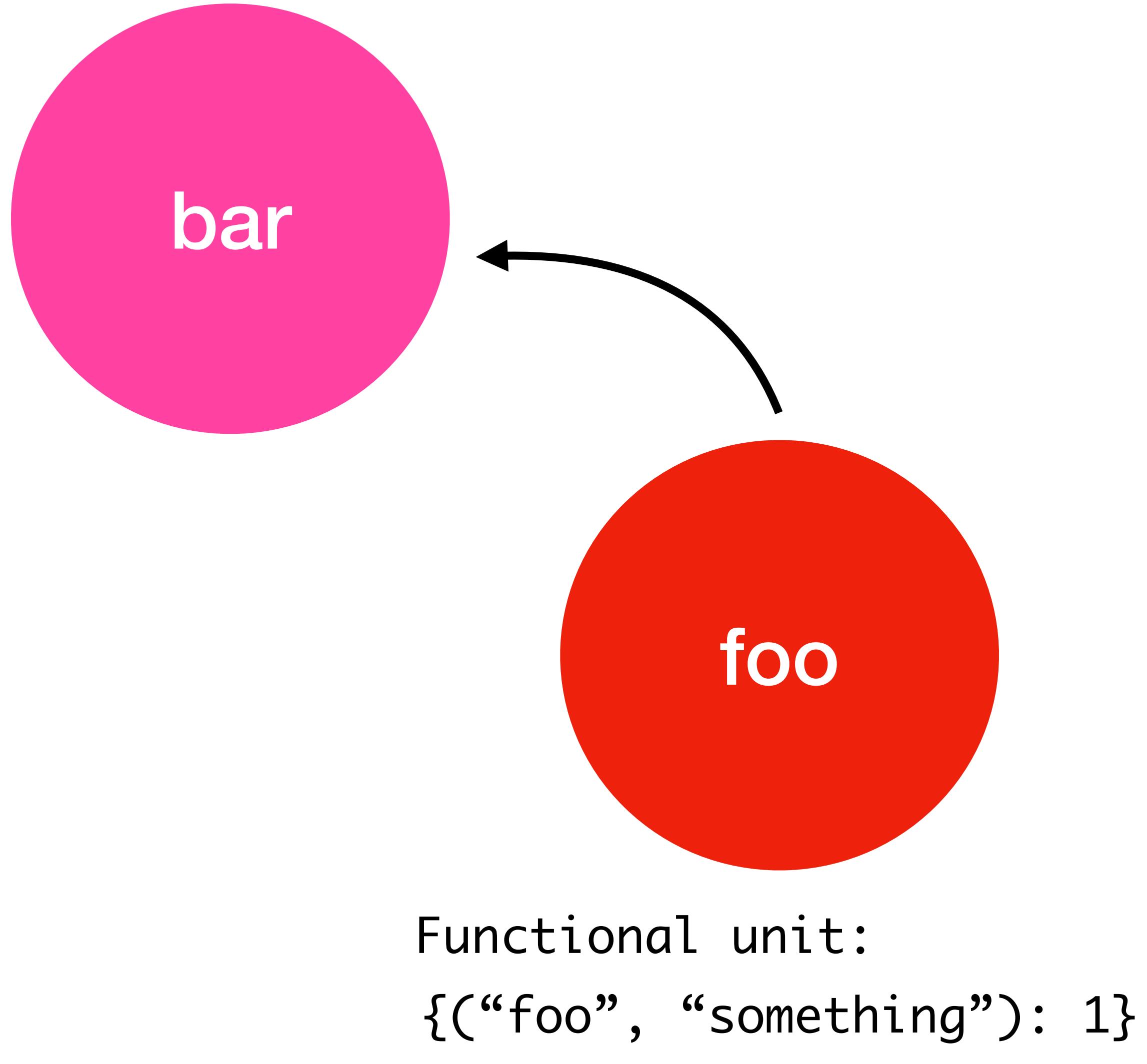
- Pass mapping data directly instead of importing it from bw2data
- Make passing mapping data optional
- Long-term: Unique integer id becomes “primary” unique identified
 - Any other identifier is backend-specific

Removing database dependency from bw2calc



Functional unit:
{"foo", "something": 1}

Removing database dependency from bw2calc



Strategy:

- Calculate dependency graph *before* passing function unit
- `prepare_lca_inputs` function

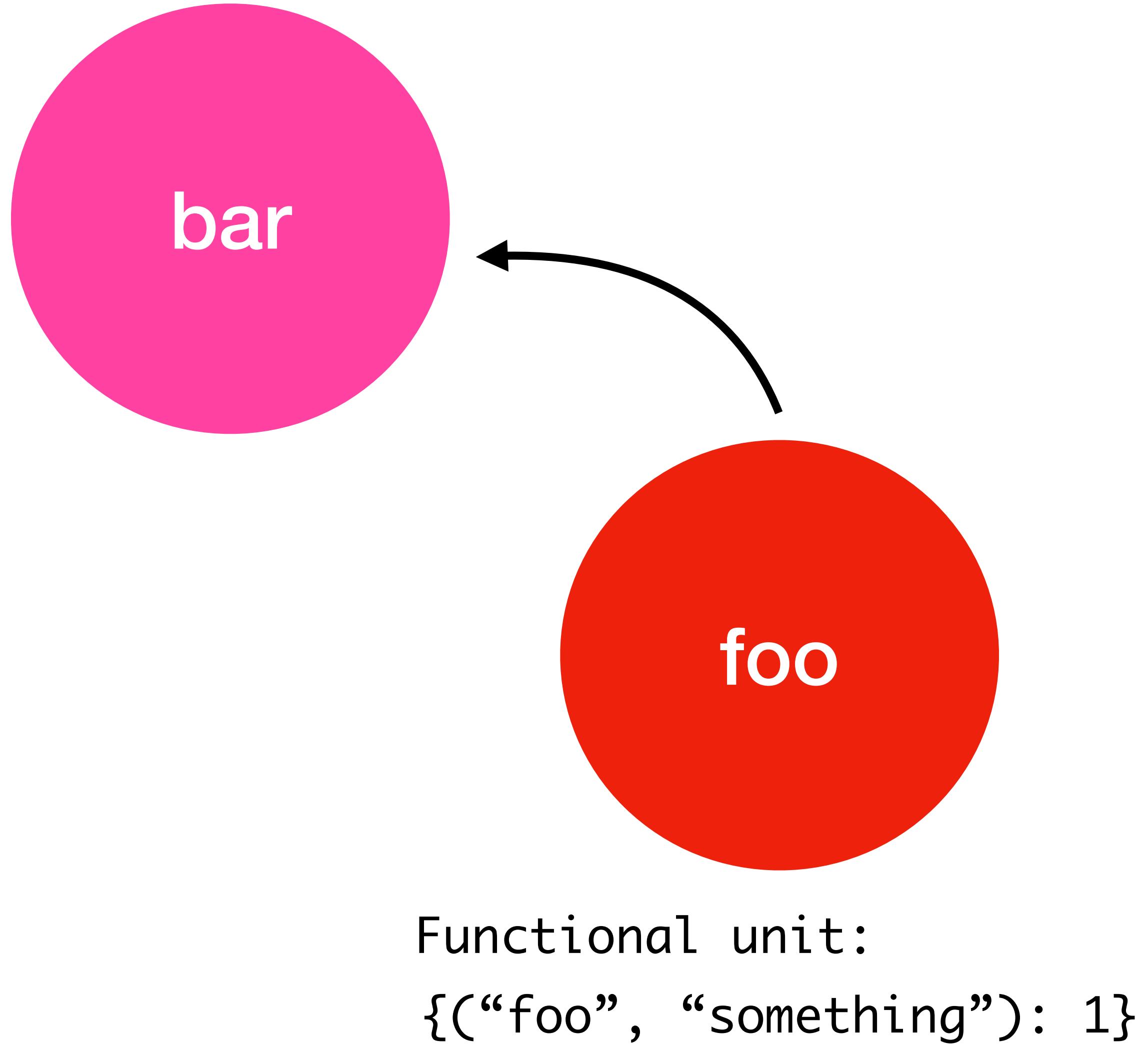
Old calling convention:

```
LCA(  
    demand={"foo": "something": 1},  
    method=("IPCC", "Climate Change")  
)
```

New calling convention:

```
LCA(prepare_lca_inputs(  
    demand={"foo": "something": 1},  
    method=("IPCC", "Climate Change")  
)
```

Removing database dependency from bw2calc



Strategy:

- Calculate dependency graph *before* passing function unit
- `prepare_lca_inputs` function

Old calling convention:

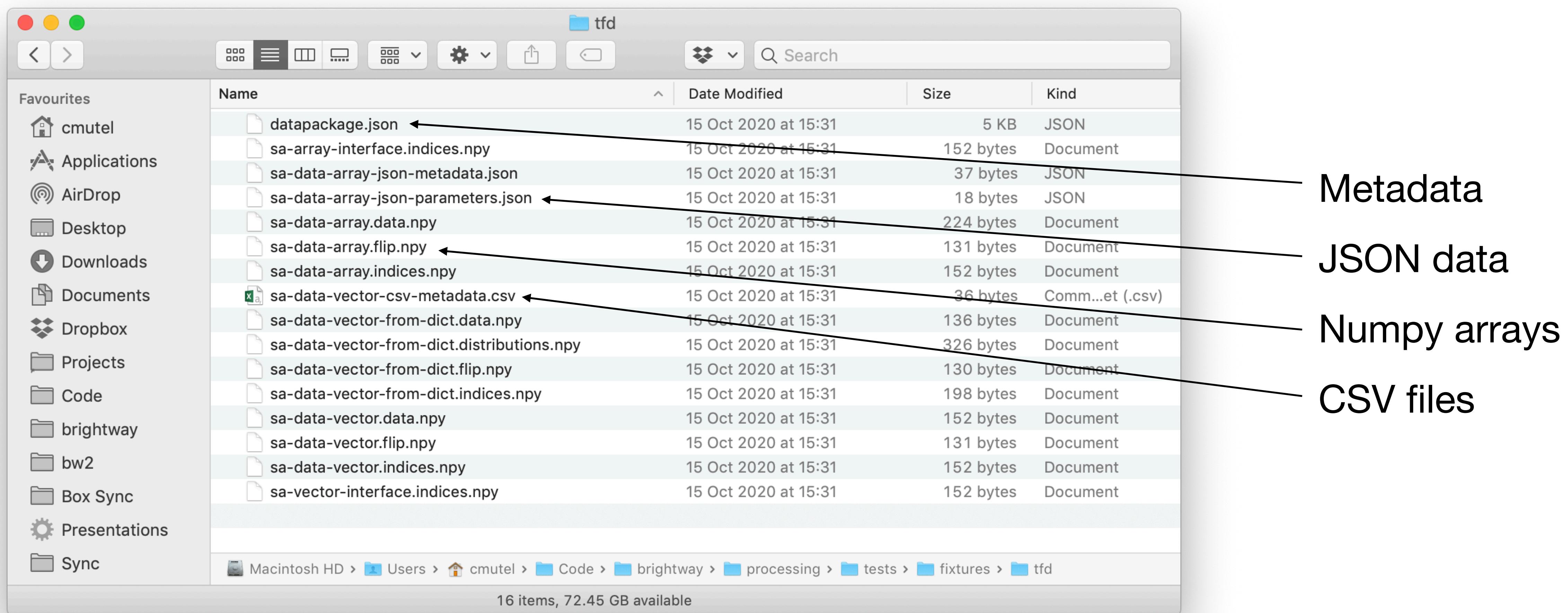
```
LCA(  
    demand={"foo", "something": 1},  
    method=("IPCC", "Climate Change")  
)
```

New calling convention:

```
LCA(  
    demand={42: 1},  
    data_objs=["some", "filepaths"]  
)
```

bw_processing Datapackages

- Based on Open Knowledge Foundation Datapackage spec
 - Same spec as presamples data packages



Processed arrays changes

input	output	row	col	type	amount	uncertainty_type	loc	shape	scale	minimum
9829	9829	4294967295	4294967295	0	1.0					
9708	9708	4294967295	4294967295	0	1.0					
9633	9633	4294967295	4294967295	0	1.0					
9276	9276	4294967295	4294967295	0	3.0999					

From mapping dict

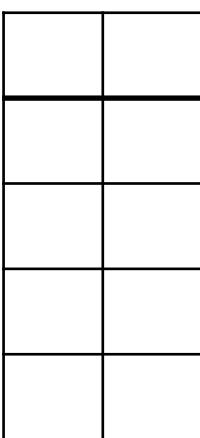
Magic

TYPE_DICTIONARY

Magic Data

Definitions of probability distribution functions

Processed arrays changes



foo.indices.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

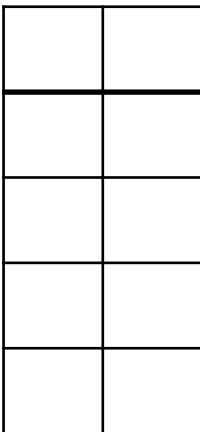
Magic

TYPE_DICTIONARY
Magic Data

uncertainty_type
loc
shape
scale
minimum

Definitions of
probability
distribution
functions

Processed arrays changes



foo.indices.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

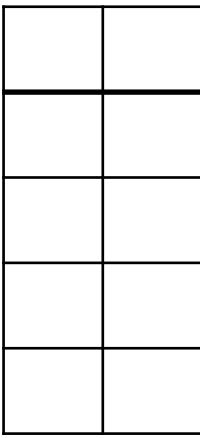
Magic

TYPE_DICTIONARY
Magic Data

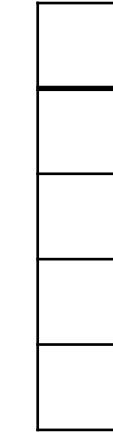
uncertainty_type
loc
shape
scale
minimum

Definitions of
probability
distribution
functions

Processed arrays changes



foo.indices.npy



foo.flip.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

Magic

TYPE_DICTIONARY
Magic Data

uncertainty_type
loc
shape
scale
minimum

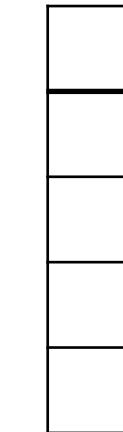
Definitions of
probability
distribution
functions

Processed arrays changes

(plus separate technosphere & biosphere packages)



foo.indices.npy



foo.flip.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

Magic

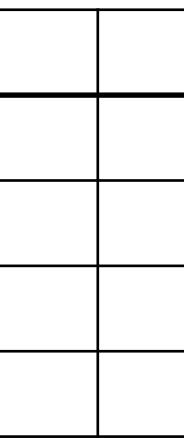
TYPE_DICTIONARY
Magic Data

uncertainty_type
loc
shape
scale
minimum

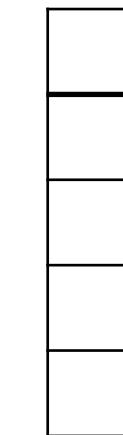
Definitions of
probability
distribution
functions

Processed arrays changes

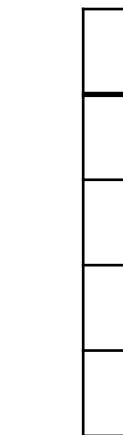
(plus separate technosphere & biosphere packages)



foo.indices.npy



foo.flip.npy



foo.data.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

Magic

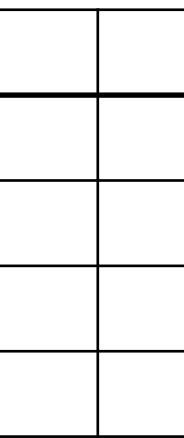
TYPE_DICTIONARY
Magic Data

uncertainty_type
loc
shape
scale
minimum

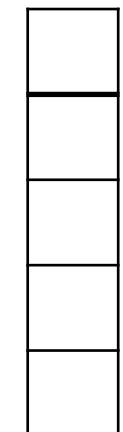
Definitions of
probability
distribution
functions

Processed arrays changes

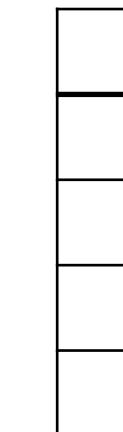
(plus separate technosphere & biosphere packages)



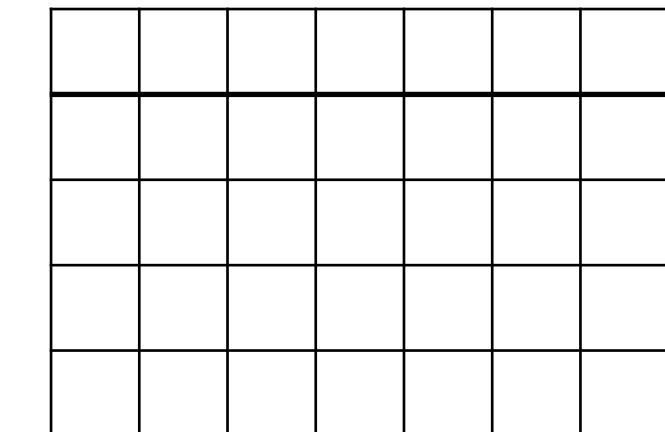
foo.indices.npy



foo.flip.npy



foo.data.npy



foo.uncertainties.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

Magic

TYPE_DICTIONARY
Magic Data

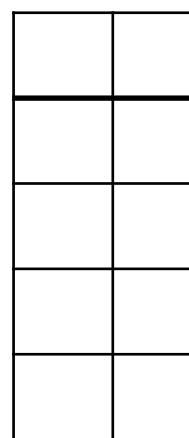
uncertainty_type
loc
shape
scale
minimum

Definitions of
probability
distribution
functions

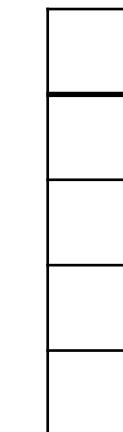
Processed arrays changes

(plus separate technosphere & biosphere packages)

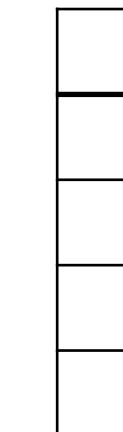
(plus numpy.Nan instead of magic values)



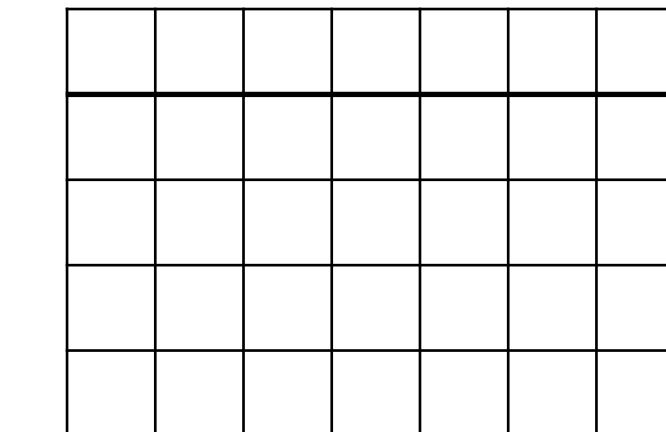
foo.indices.npy



foo.flip.npy



foo.data.npy



foo.uncertainties.npy

input	output	row	col	type	amount
9829	9829	4294967295	4294967295	0	1.0
9708	9708	4294967295	4294967295	0	1.0
9633	9633	4294967295	4294967295	0	1.0
9276	9276	4294967295	4294967295	0	3.0999

From
mapping
dict

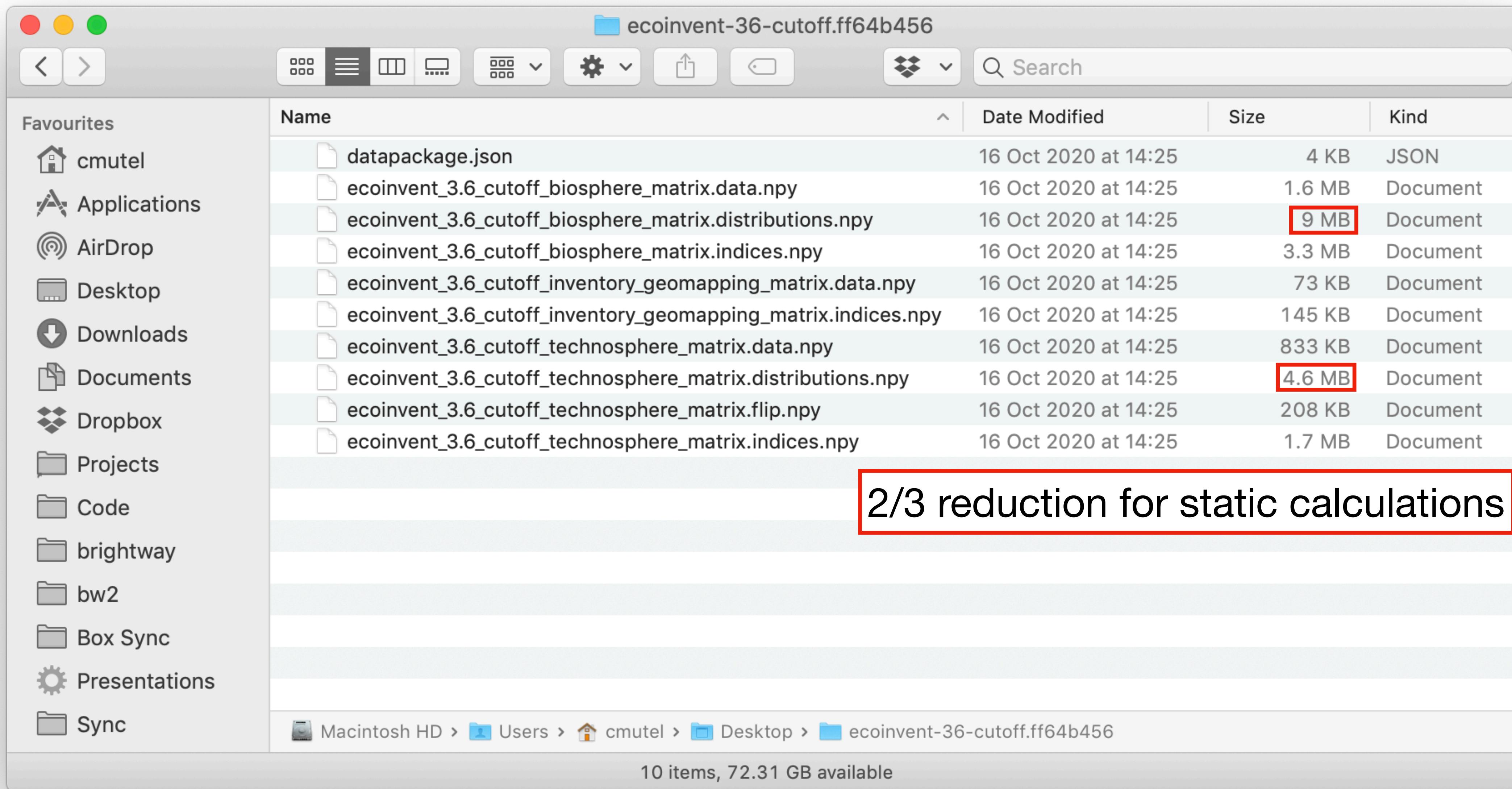
Magic

TYPE_DICTIONARY
Magic Data

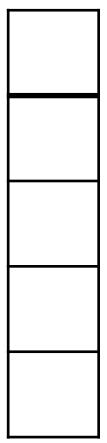
uncertainty_type
loc
shape
scale
minimum

Definitions of
probability
distribution
functions

1. Reduce memory and IO



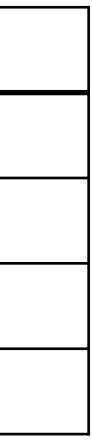
2. Alternative uncertainty models



`foo.data.npy`

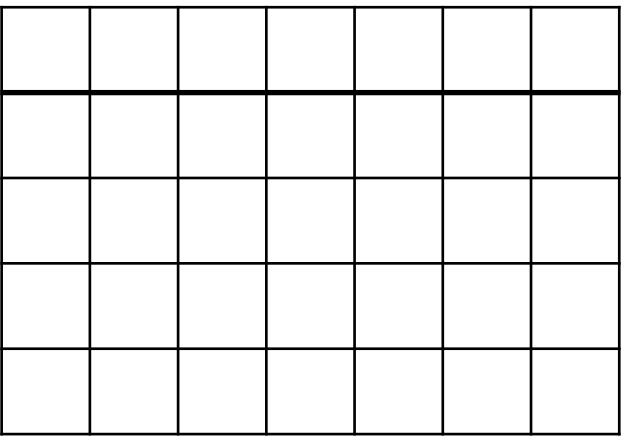
Vector

2. Alternative uncertainty models



`foo.data.npy`

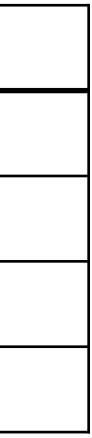
Vector



`bar.data.npy`

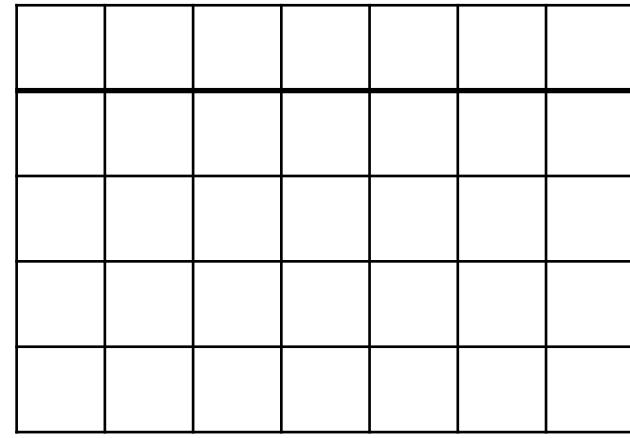
Array

2. Alternative uncertainty models



`foo.data.npy`

Vector



`bar.data.npy`

Array

Concept of presamples

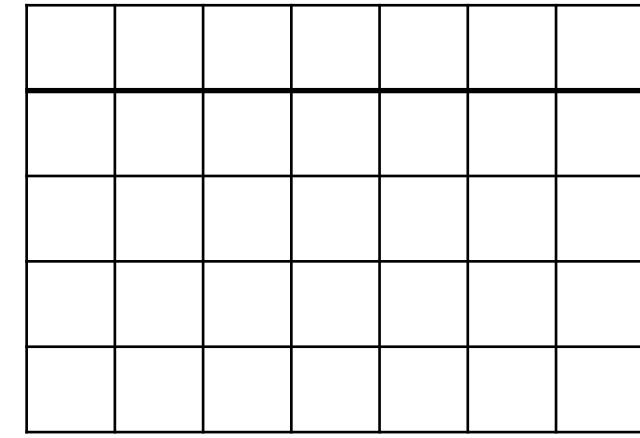
- Alternative system configurations
- Choices among market providers
- Correlation within or across activities
- Correlation across time and space (LCI & LCIA)
- Result from external model
- Sampling population data (e.g. time series)

2. Alternative uncertainty models



`foo.data.npy`

Vector



`bar.data.npy`

Array

One serialisation format

One interface

Same `indices.npy` and `flip.npy`

Concept of presamples

- Alternative system configurations
- Choices among market providers
- Correlation within or across activities
- Correlation across time and space (LCI & LCIA)
- Result from external model
- Sampling population data (e.g. time series)

3. Static versus dynamic resources

Static resources can be saved to files

Dynamic resources are not saved

- Resolved when matrices are built

Dynamic resource use case:

- Data from another program
- Provided by a web service
- Data that comes from generators
- Manipulation before matrix insertion

3. Static versus dynamic resources

Static resources can be saved to files

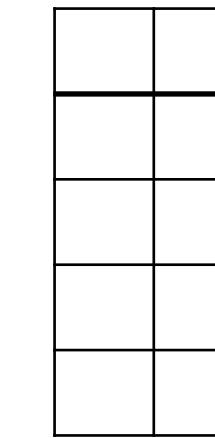
Dynamic resources are not saved

- Resolved when matrices are built

Dynamic resource use case:

- Data from another program
- Provided by a web service
- Data that comes from an generators
- Manipulation before matrix insertion

Serializing dynamic interfaces:



`foo.indices.npy`



`foo.flip.npy`

Structure of dynamic data is fixed.

Need to create a Python object that supports `next()` *after* the Datapackage is loaded.

3. Static versus dynamic resources

Static resources can be saved to files

Dynamic resources are not saved

- Resolved when matrices are built

Dynamic resource use case:

- Data from another program
- Provided by a web service
- Data that comes from an generators
- Manipulation before matrix insertion

Datapackages come in 3 flavours:



Single zip file
Easily portable

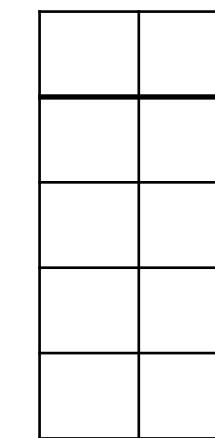


Directory
Transparent



In-memory
Flexible

Serializing dynamic interfaces:



`foo.indices.npy`



`foo.flip.npy`

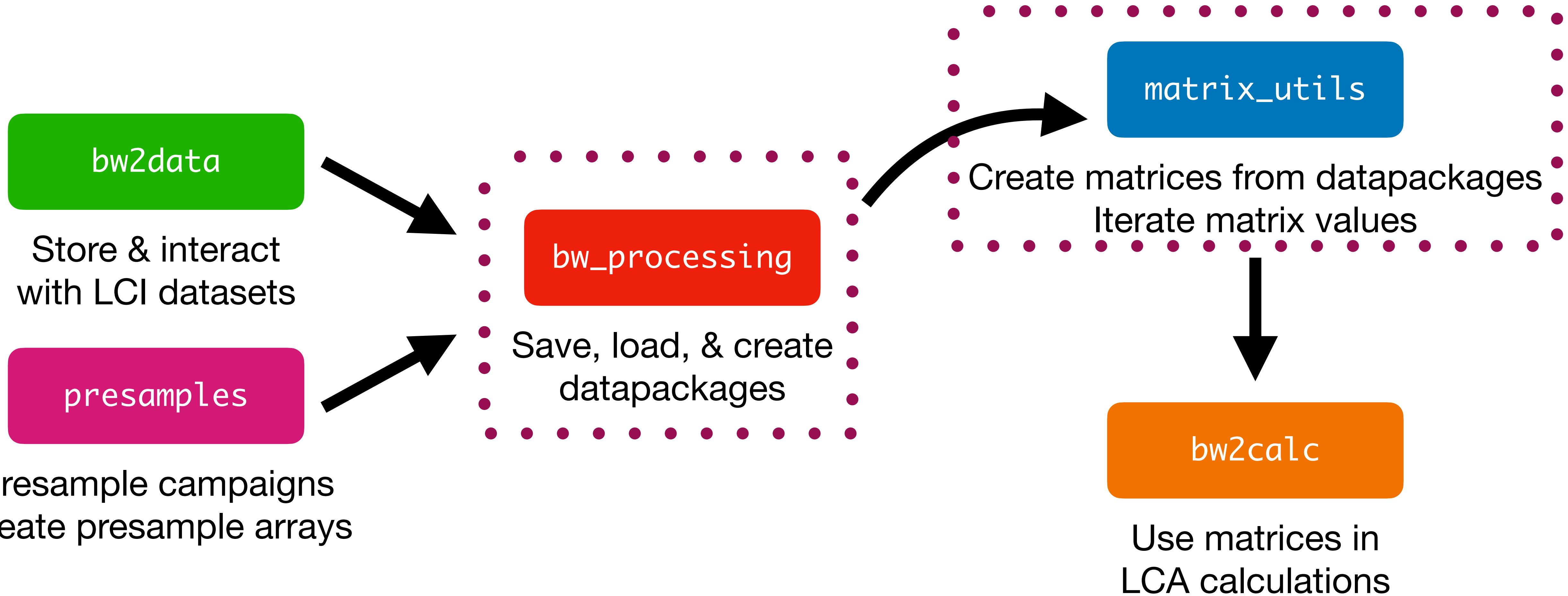
Structure of dynamic data is fixed.

Need to create a Python object that supports `next()` after the Datapackage is loaded.

4. Cloud computing

- No need for bw2data or installed databases
- Datapackages can be zip files
- Web resources can be dynamic resources

New libraries



Release schedule



Datapackage format is fixed & can be used (see [bw_processing](#))

Testing release on **Nov. 2**

General availability release on **Dec. 14**

Photo source