bw2-regional Documentation

Release 0.2

Chris Mutel

May 06, 2015

It is easy to do regionalized LCA incorrectly. This package tries to make it at least a bit easier to do regionalization correctly, for at least some definition of correctly.

bw2regional supports the following regionalization calculations:

1. Inventory database and impact assessment method share the same spatial scale (*Shared spatial scale*)

2. Inventory and impact assessment have different spatial scales (*Two spatial scales*)

3. Inventory and impact assessment have different spatial scales, with background loading used for spatial allocation (*Two spatial scales with generic loading*)

In addition to making regionalized LCA calculations, maps of regionalized impact can be exported from methods 2 and 3.

bw2regional is designed to work closely with pandarus, a library for matching spatial data sets and calculating their mutual intersected areas.

bw2regional introduces the idea of *Geocollections*, which organize and identify spatial scales, similar to how Databases are used in bw2data.

A number of utility functions should make importing and manipulating the spatial data needed.

Contents:

Data and data formats

1.1 Geocollections

Geocollections are containers that organize and describe sets of spatial data identifiers, and possibly other data, including spatial supports. They are similar to the concept of Databases in Brightway2: an inventory dataset could be identified by (`"My new database"`, `"Dataset 14"`), while spatial unit could have a similar id: (`"My new raster"`, `"Cell 42, 11"`). However, only metadata is stored for each geocollection, so there is no `Geocollection` object, only `geocollections`.

Geocollections are used by both inventory data sets (e.g. for custom locations) and impact assessment methods.

Geocollections can refer to vector or raster data. For example, one geocollection could be the set of world countries, as described by the Natural Earth data, while another could be the raster cells used in a particular impact assessment method.

Geocollections have two purposes in Brightway2:

- They provide a conceptual and physical grouping of spatial data into manageable units
- They provide a way to uniquely identify spatial data

A geocollection is not a geodatabase - no spatial data is required. However, if the original spatial data is available (as a vector or raster file), it can be specified and used later in analysis.

A geocollection is specified by a unique string, which is usually the name of the geocollection.

1.1.1 Metadata fields

There are no required metadata fields for geocollections, but some common fields are useful when the original data is available. All the below listed metadata field values should be strings:

- **filepath**: Filepath for the vector or raster file
- **layer**: Layer name (vector data only)
- **field**: Field name that uniquely identifies each feature (vector data only)
- **vfs**: Virtual file system used to load zipped shapefiles
- **encoding**: Text encoding

Note: It is *strongly* preferred that ESRI shapefile be zipped to a single file, with appropriate metadata. Ideally, each Geocollection can have its associated spatial data in a single file. For example, the Natural Earth 50m political data is specified as:

```
vfs="zip:///Users/cmutel/Downloads/Geodata/ne_50m_admin_0_countries.zip",
filepath="/ne_50m_admin_0_countries.shp.
```
See [the Fiona manual](#) for more.

Note: Data specific to a given impact assessment method, like which band has the characterization factors for a given impact category, should be stored in the impact assessment metadata.

1.1.2 Standard geocollections

The following are some standard geocollections, and are installed automatically.

- **global**: Only the global location "GLO"
- **countries**: All the countries in the world, as defined by ISO, and identified by the ISO 2-letter codes.
- **regions**: [UN regions and subregions](#)
- **ecoinvent 2**: Special regions defined by ecoinvent in version 2 of the database
- **ecoinvent 3**: Special regions defined by ecoinvent in version 3 of the database

1.2 Regionalized impact assessment

Regionalized impact assessment methods have characterization factors that depend both on biosphere flows and spatial units. Characterization factors can be static or uncertain.

In Brightway2, the `Method` object can store either site-generic or regionalized IA methods. Similarly, the `methods` metadata store works the same for site-generic and regionalized IA methods.

1.2.1 Specifying spatial objects

Following the Brightway2 principle of KISS, spatial objects have a simple format with reasonable defaults:

1. If not location data is provided, the global location "GLO" is assumed
2. A two-letter ISO county code can be used
3. Otherwise, spatial objects should be identified by the combination of geocollection and id, just like objects in Databases: `(geocollection, spatial unit id)`.

The utility function *Normalize spatial units* should be used to normalize options 1 and 2 with the respective geocollections.

1.2.2 Metadata

In addition to the standard metadata for IA methods, such as `unit` and `description`, regionalized IA methods should include the following for complete functionality:

- **geocollections**: List of one or more strings identifying the geocollections. Normally only one geocollection is associated with a regionalized IA method. See *Geocollections*.
- **band**: Band number in original raster data set. Needed to import characterization factors.
- **cf_field**: Field name of characterization factor value. Needed to import characterization factors.

1.2.3 Data format

Site-generic IA methods have a simple data format:

```
[
    [biosphere flow, maybe uncertainty],
]
```

Where `maybe uncertainty` is either a floating point number (implying no uncertainty), or a stats_array uncertainty dictionary, like:

```
{'loc': 2, 'scale': 0.5, 'uncertainty_type': NormalUncertainty.id}
```

Note: In site-generic CFs, where a location is not given, the "GLO" location is assumed.

Regionalized IA methods are almost the same; they just have an additional field for a location specifier.

```
[
    [biosphere flow, maybe uncertainty, spatial object],
]
```

1.3 Loadings

Background loadings are data on the spatial patterns of emission, and are used to determine the relatively likelihood that a given inventory dataset occurs in a impact assessment spatial unit. The idea is that the existing patterns of emissions are reasonable predictors of where present or future emissions will occur.

In Brightway2, background loadings are represented by the `Loading` objects, and metadata about all loadings is stored in `loadings`.

Because loadings are a density of predicted activity, their unit is physical quantity (e.g. mass or energy) per unit area. Loadings are multiplied by intersected areas, and then normalized by total loading, so their units are canceled out in the end.

As loadings are emission-specific, this could mean that different loadings for different biosphere flows could predict different spatial patterns of inventory activity. There is no real research on the importance of this inconsistency.

Background loadings are, in general, supplied by the impact assessment method developers. If no loadings are supplied, the generic fallback is to allocate impact assessment spatial units to inventory spatial unit by intersected area; however, proxy loadings can be independently calculated. See **this example ipython notebook** for one such procedure.

Background loadings should always use the same geocollection as their IA methods.

1.3.1 Metadata

Their are no required fields for background loadings, as most metadata comes from the IA method. Fields such as `description` can be used.

1.3.2 Data format

```
[
    [maybe_uncertainty, location_id],
]
```

> **Warning:** Industry sector-specific loadings and emission-specific loadings are not yet supported in bw2regional.

1.4 Intersections

Data for the geographic transform matrix **G** is stored in Intersection objects. In Brightway2, areal intersection data is represented by the `Intersection` object, and the metadata store is `intersections`. Each combination of geocollections should be a separate `Intersection` object.

Intersection data is calculated using pandarus. Conversion from the pandarus data format is done with the utility function *Load Pandarus output*.

1.4.1 Metadata

No metadata is required for intersections.

The `pandarus-filepath` field is the filepath of the Pandarus file, and is set automatically when imported.

1.4.2 Data format

We assume that data is written automatically after conversion from pandarus, so users shouldn't be writing or manipulating intersection data themselves. Nevertheless, here is the data format:

```
[
    [spatial object 1, spatial object 2, intersection area],
]
```

1.5 Extension tables

1.5.1 Metadata

In addition to the standard metadata for IA methods, such as `unit` and `description`, regionalized IA methods should include the following for complete functionality:

- **geocollection**: String identifying a geocollection. See *Geocollections*.
- **xt_field**: Field name used for extension table values. Only needed for vector spatial data.
- **band**: Raster band index for extension table values.

1.5.2 Data format

```
[
    [float, spatial object 1]
]
```

Understanding Regionalized LCA

2.1 Two Spatial Scales with Generic Loading

We start with a relatively complex example of regionalized LCA - the inventory database and impact assessment method have different spatial scales, and we have background loading data which is generic to all biosphere flows. In this case, we have the following formula:

$$h_r = [\mathbf{MNGLR}]^T \circ [\mathbf{B} \cdot (\mathbf{A}^{-1}f)]$$

Let's start with $\mathbf{R}$, the regionalized characterization matrix. $\mathbf{R}$ provides characterization factors for each biosphere flow and each impact assessment spatial unit. The spatial scale for impact assessment depends on the impact category, but it generally something like watersheds or ecoregions.

$\mathbf{R}$ has rows of different impact assessment spatial units, and columns of biosphere flows. In our example, we are using made up numbers, so we can make up $\mathbf{R}$ as follows:

$$\mathbf{R} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

So, in the first row (perhaps the Danube watershed) and the first column (perhaps emission of elemental chlorine), there is a characterization factor of 1. The units in $\mathbf{R}$ are units of damage, either at the mid- or endpoint.

We now start estimating where our inventory activities occur. Of course, we know where they occur, at least in our version of the world - each inventory dataset should have a location, and the mapping matrix $\mathbf{M}$ indicates which inventory dataset is associated with inventory location. The number of inventory locations is always equal to or fewer than the number of inventory datasets, as any additional places on earth which we don't have inventory datasets for don't exist in our model of the world. $\mathbf{M}$ has rows of inventory activities (e.g. make steel), and columns of inventory spatial units (e.g. Georgia). If the activity occurs in the given location, there is a one in $\mathbf{M}$, and otherwise a zero. $\mathbf{M}$ is unitless.

Here is our example mapping matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This $\mathbf{M}$ tells us that the activity in the first row, whatever that is, happens in the first inventory location, wherever that is.

So we know where our inventory activities are in the spatial scale of the inventory, but not in the spatial scale of the impact assessment. To apply the characterization factors in the spatial scale of the impact assessment to the calculated inventory, we need to match the two scales.

We can used a GIS, or more simply, the pandarus utility, to calculate the intersected areas of each inventory spatial unit with each impact assessment spatial unit. This is the first step in figuring out where activities occur in the spatial scale of the inventory - we could allocate based on intersected area. However, area is a pretty poor proxy for where industrial activities actually occur, so we will need additional information.

The matrix **G** gives us the intersected areas for the two spatial scales. For example, it could tell us that the Rhone watershed has a large amount of area in the borders of France, but none in Brazil. **G** has rows of inventory spatial units, like countries or political unions, and columns or impact assessment spatial units, like watersheds or ecoregions. **G** has units of area, by convention square meters in brightway2 and pandarus.

Our example **G** is:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 0 & 5 & 8 \\ 0 & 0 & 13 \end{bmatrix}$$

In this example matrix, ...

It now gets a bit tricky to think about, because we are going to use environmental data to estimate industrial activity, and we mostly go the other direction in LCA. We want to know how likely it is that industrial activity, or more generally, activity which emits or consumes resources, occurs in each impact assessment region. We can't just use the area of the impact assessment region, because the world's economic activity is not uniformly distributed in space. There are more activities in the Rhein watershed than the Lena river watershed, for example. We can use databases of existing environmental loadings to estimate where activities and their emissions are taking place.

The background loading matrix **L** represents our best knowledge about where inventory activities happen, based on where these activities are happening now and the amount that is currently emitted to the environment. **L** is diagonal - it has values on the diagonal and zeros elsewhere.

Where the normalization matrix **N** is defined by:

$$\mathbf{N}_{i,i} = \left[\sum_j (\mathbf{GL})_{i,j} \right]^{-1}$$

Common tasks with regionalized data

3.1 Match geocollections using Pandarus

3.2 Normalize spatial units

3.3 Convert geocollection to Method data

Note: Conversion requires access to the actual spatial data and all needed metadata, i.e. filepaths, layer and field name for shapefiles, and raster band for rasters.

3.4 Convert geocollection to Loading data

3.5 Load Pandarus output

To load the output from a Pandarus calculation, use `Pandarus.import_from_pandarus` (insert ref. to method docs here) like:

```
Intersection(("from geocollection name", "to geocollection name")).import_from_pandarus("")
```

Note: Be sure to pass the `--lca` if calling Pandarus from the command line.

Warning: Make sure the order of the two geocollections is the same in the Intersection object as it was in the Pandarus calculation.

LCA classes - technical reference

4.1 Regionalization mixin

4.2 Two spatial scales with generic loading

4.3 Two spatial scales

4.4 Shared spatial scale

4.5 Extension tables

Development

bw2regional is developed by Chris Mutel as part of his work at the Ecological Systems Design group at ETH Zürich. Source code is available on bitbucket.

Indices and tables

- genindex

- modindex

- search