Structured Publication of Life Cycle Assessment Models and Results

Brandon Kuczenski
Institute for Social, Behavioral, and Economic Research
University of California, Santa Barbara

2017-03-31

Supporting Information

1 Interpreting the Research Object

The product systems are provided as "research objects" that take the form of Excel spreadsheets. The spreadsheets are provided in the supplementary information accompanying the paper. Each example is also illustrated with a table in the body of the paper. The spreadsheets allow a data user to interpret, validate, and re-create the foreground study. Each spreadsheet contains an entity map, a table of LCIA category scores, a set of data matrices, and a set of aggregation results. The aggregation results can be computed from the data tables, and the LCIA results can be computed from the aggregation results.

1.1 The Entity Map

The "EntityMap" tab describes each entity in the model, broken into five sections: characterization quantities (for LCIA); foreground nodes; background dependencies; cutoff flows; and elementary flows. Each entity is given an identifying key that is used to represent the entity in the data tables.

In addition to the key, each entity has four core properties:

- The *origin* defines the data resource from which the entity was derived. In an ideal case, the origin would be a URI of a namespace on the Semantic Web. Since the research object is itself the data resource that defines the foreground, the origin for foreground nodes is simply "Foreground."
- The *identifier* is unique with respect to the origin. The identifier should allow a data user to find the entity's dataset when consulting the origin's namespace. Ideally, the origin and the identifier together form a URI for the entity. In principle a data user should be able to retrieve the correct entity using only the origin and the identifier fields.
- Each entity has a *reference unit* that defines how numeric values should be interpreted.
- Each entity also has a *name* for convenient interpretation of the entity.

Each entity type (aside from characterization quantities) has additional descriptive fields:

- The *flow direction* describes the direction of each flow with respect to the entity being defined. Foreground flows, background dependencies, cutoff flows and elementary flows all have directions.
- Foreground nodes also have a *flow name* that reports the name of the reference flow emanating from the node.
- Background dependencies also have a *reference flow* that tells which of the process's reference outputs is being used. In the case of multi-output background processes, this field is necessary to identify the appropriate allocation result.
- Elementary flows also have a *compartment* that describes which environmental compartment the flow is exchanged with. This is used to determine the correct characterization factor.

Note that cutoff flows and elementary flows are both included in the foreground emission matrix, B_f, because they are mathematically equivalent. In other words, "emissions" can be interpreted to include any flows that are exterior to the technology matrix, including resource withdrawals, environmental releases, and cutoffs.

1.2 LCIA Scores

The “LciaScores” tab has one data column for each LCIA method included in the entity map. The rows of data include:

- `s_tilde`, for $\tilde{s}$, the total LCIA result for the product system;
- `sf_tilde`, for $\tilde{s}_f = \mathbf{e} \times B_f \times \tilde{\mathbf{x}}$, the sum of characterized foreground emissions;
- `sx_tilde`, for $\tilde{s}_x = \mathbf{e} \times B_x \times A_d \times \tilde{\mathbf{x}}$, the sum of characterized background emissions;
- One row for each background dependency, reporting the unit impact score of the named dependency. These values can be validated by consulting the original LCI database provider.

1.3 Data Sheets

After the entity map and the LCIA results, the spreadsheet includes worksheets that report the contents of the E, A_f, A_d, and B_f matrices, as well as the aggregation results $\tilde{\mathbf{x}}$, $\tilde{\mathbf{a}}_d$, and $\tilde{\mathbf{b}}_f$. The aggregation results can be validated from the data tables using Eqs. 6 and 8 in the main paper. The data sheets can be presented in sparse format, which contains three columns: a row key, a column key, and a data value, where the row and column keys each correspond to an entity on the entity map. Alternately, the data sheets can be presented in full format, in which the top row contains a list of column keys, the first column contains a list of row keys, and the data region include nonzero values at the intersection of the corresponding row and column keys (zero valued cells are left blank.)

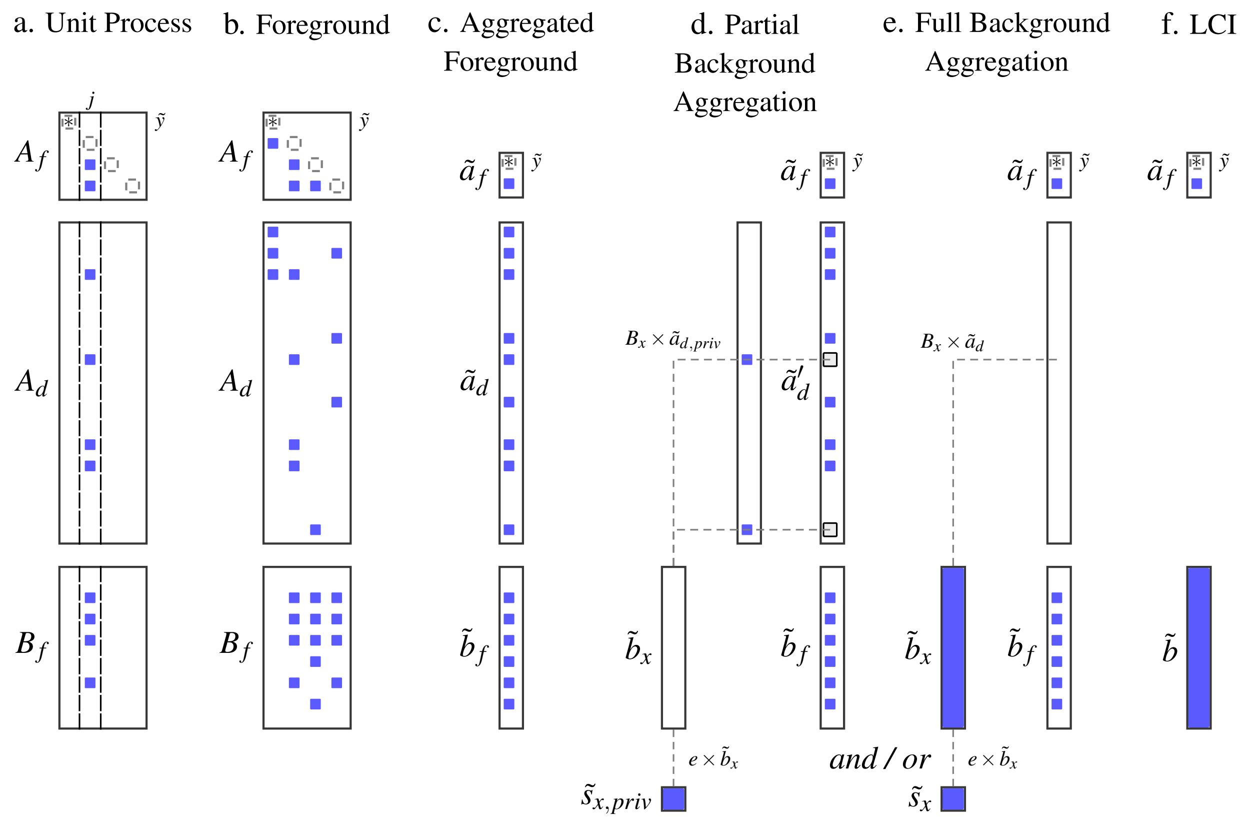

Figure S1: Publication structures for different forms of foreground aggregation.

2 Aggregation of Foreground Models

This section describes what data points must be reported in order to generate a structured publication of a foreground model in various stages of aggregation. In each case, full publication must include the two components already introduced: the entity map, used to and enumerate the matrix rows and columns; and the tabular data, used to populate the matrices. For complete fidelity to the model, each non-zero entry in any matrix should be represented in the tabular data. If results are included, the transparency of the publication can be improved by reporting the characterization vector as well. Note that the reference flows of each foreground node are implicit in Eq. 6 and should not be replicated in the tabular data. Different aggregation forms are illustrated in Figure S1.

2.1 Unit Process Inventory

A unit process makes up a single shared column of A_f, A_d, and B_f, as depicted in Figure S1(a). Publication formats for unit processes already exist, the most well-known being the ILCD and Ecospold XML formats. Current LCA software is based on the principle of the unit process as the sole consensus format for structured publication, although current formats do not uniformly distinguish between foreground flows, dependencies, and emissions. The entity map must include at least one foreground flow, which is the reference flow of the process, and corresponds to the column of A_f to which it belongs. Multi-functional processes can also be expressed, but a reference exchange must still be selected before the process can be entered into a foreground matrix. Other co-products should in this case be listed as additional foreground flows in the entity map, and their exchange values included in the tabulated data; absent further specification, these flows would be interpreted as cutoff flows.

2.2 Foreground

See Figure S1(b). A foreground can be viewed as a linked set of unit processes. In the mode of structured publication as a research object, the entries in the unit process inventories must be segregated into foreground exchanges (entries in A_f), background dependencies (entries in A_d), and emissions (entries in B_f). A foreground or foreground fragment comprises these three complete matrices, all having a common set of columns. A publication in this vein would allow a data user to completely reproduce the study fragment. A foreground fragment is also the smallest publication unit that can fully express an allocation treatment, because such treatment requires the involvement of multiple unit processes, either for the purposes of partitioning, co-product substitution, or system expansion. The two examples presented above show foreground publications.

2.3 Aggregated Foreground

See Figure S1(c). Any collection of foreground nodes can also be expressed in aggregated form by performing a row-wise weighted sum of the foreground columns. This is equivalent to multiplying the A_d and B_f matrices by $\tilde{\mathbf{x}}$ to yield $\tilde{\mathbf{a}}_d$ and $\tilde{\mathbf{b}}_f$. The foreground vector $\tilde{\mathbf{a}}_f$ is obtained by selecting all the elements of $\tilde{\mathbf{x}} - \tilde{\mathbf{y}}$ that represent input, output, or cutoff flows. A publication of the aggregated foreground is identical to the publication of a unit process, except that the result was derived through aggregation. As in the unit process case, the reference flow should appear in the entity map but the reference output should not be reported in the tabular data.

2.4 Partial Background Aggregation

See Figure S1(d). This approach is useful when an author wishes not to disclose certain dependency relationships in the foreground but still wants to publish a complete, reproducible model. The aggregated dependency vector is split into two parts that sum to the original:

$$\tilde{\mathbf{a}}_d = \tilde{\mathbf{a}}_{d,priv} + \tilde{\mathbf{a}}'_d \tag{1}$$

The disclosed dependencies $\tilde{\mathbf{a}}'_d$ are reported, and the private dependencies $\tilde{\mathbf{a}}_{d,priv}$ are replaced with a private impact score $\tilde{s}_{x,priv}$, computed from corresponding life cycle inventories from the background database. Foreground emissions can still be reported separately, or they can be characterized and combined with the private score as appropriate. This approach allows the author to limit the disclosure of confidential information; however, the reader will still learn how large of an impact can be attributed to the undisclosed portion. It isimpossible to discern the makeup of the private dependency vector from the private impact score.

2.5 Full Background Aggregation

See Figure S1(e). In this approach, all dependency information is concealed through replacement with an impact score derived from the background life cycle inventory. In this publication, foreground emissions are still reported separately, along with any foreground inputs, outputs, and cutoffs. For a marginal increase in transparency, the full background inventory can be reported as $\tilde{\mathbf{b}}_x$. However, taking this approach significantly increases the size of the publication because the background inventory vector is full (not sparse). Full background aggregation is most commonly practiced

2.6 Full LCI

See Figure S1(f). Here, all foreground information is concealed and only the aggregated life cycle inventory vector $\tilde{\mathbf{b}}$ is reported. The life cycle inventory $\tilde{\mathbf{b}}$ provides the most aggregated form of the study that can still be independently validated with an external characterization vector $\mathbf{e}$.

3 LCIA Case Study: Marine Eutrophication

In the Organic Potatoes example presented in the text, the product system was characterized with respect to six different LCIA methods, all representing different implementations of the ReCiPe Marine Eutrophication midpoint indicator, measured in kg N-equivalents. Ecoinvent's LCIA implementation is available by request from the Ecoinvent Centre; it has not changed since 2014. The ReCiPe method has been implemented according to the Individualist (I), Hierarchist (H), and Egalitarian (E) modes specified in the methodology; in addition, the (E) and (H) methods have been reimplemented in a way that omits long-term emissions. The European Life Cycle Database's reference implementation of marine eutrophication was last updated in 2011. The dataset cites the

2008 version of ReCiPe.

The tabular output of the structured publication provides a convenient method to compare the implementations. Inspection reveals that, as far as marine eutrophication is concerned, there is no difference among the (I), (H), and (E) modes, which all have the same characterization factors. Similarly, the (E) and (H) methods omitting long-term emissions are also the same. Both sets of Ecoinvent indicators bear similarities and differences to the ILCD implementation, however. The differences are shown in Table S1.

Briefly, the implementations correspond on many significant flows but differ on several as well.

- While all methods characterize ammonia to air and ammonium ion to water, only ILCD characterizes ammonia to water and ammonium to air.

Table S1: ReCiPe Marine Eutrophication, Midpoint, kg N-equivalent

Flowable	Compartment		Ecoinvent w/o LT	Ecoinvent LT	ILCD Matching	ILCD Non-matching
Ammonia	air	all	0.092	0.092	0.092	
Ammonium, ion	water	ocean	1E-200	0.778	0.778	
Ammonium, ion	water	all other	0.78	0.778	0.778	
Cyanide	water	ocean	1E-200			
Cyanide	water	all other	0.54			
Nitrate	air	all	0.028	0.028	0.028	
Nitrate	water	long-term		0.226	0.226	
Nitrate	water	ocean	1E-200	0.226	0.226	
Nitrate	water	all other	0.23	0.226	0.226	
Nitrite	water	long-term			0.304	
Nitrite	water	ocean	1E-200		0.304	
Nitrite	water	all other	0.3		0.304	
Nitrogen	water	ocean	1E-200	1		
Nitrogen	water	all other	1	1		
Nitrogen, organic bound	water	ocean	1E-200	1		
Nitrogen, organic bound	water	all other	1	1		
Nitrogen oxides	air	all	0.039	0.389		
Ammonia	water	all				0.824
Ammonium, ion	air	all				0.087
Nitrogen, total (excluding N2)	water	all				1
Nitrogen dioxide	air	all				0.389
Nitrogen monoxide	air	all				0.596

- Where Ecoinvent characterizes “nitrogen oxides”, ILCD separately characterizes nitrogen dioxide and nitrogen monoxide. Since neither are present in the Ecoinvent data sets, the ILCD method will miss these flows entirely.
- Ecoinvent includes characterizations for two different forms of elemental nitrogen, but ILCD is characterized with respect to a third flow, which is likely a synonym. As a consequence, the ILCD implementation will not capture elemental nitrogen emissions from Ecoinvent datasets.
- Cyanide and nitrite emissions are only included in “without long-term”, and do not appear in the “long-term” emissions. Cyanide emissions do not appear at all in ILCD.
- The characterization for nitrogen oxides in “without long-term” is a factor of 10 smaller than it is in “long term.”

This example demonstrates how independent reimplementation of LCIA methods can produce inaccurate and/or inconsistent results.

4 Open Access to Paper Materials

The software used to generate the examples published in this paper is available online. Please consult the GitHub repository found at http://github.com/bkuczenski/Publication-JIE. The python code used to produce the examples is found in the ‘python’ directory, as a set of Jupyter notebooks. Because the Ecoinvent database cannot be distributed due to licensing restrictions, only the US LCI example works “out of the box.”

Please contact the author with any questions about the code, via the GitHub page or via email at `bkuczenski@ucsb.edu`.