Journal of
Industrial Ecology

Disclosure of Product System Models in Life Cycle Assessment: Achieving Transparency and Privacy

Journal:	*Journal of Industrial Ecology*
Manuscript ID	17-JIE-5822.R2
Wiley - Manuscript type:	Research & Analysis
Date Submitted by the Author:	23-Apr-2018
Complete List of Authors:	Kuczenski, Brandon; University of California, Santa Barbara, Institute for Social, Behavioral, and Economic Research
Keywords:	life cycle assessment (LCA), transparency
User-Supplied Keywords:	critical review, product system models, confidential information

Abstract:	Many of the challenges facing knowledge synthesis from life cycle assessment (LCA) studies stem from the inability of study authors and readers to formally agree on the structure and content of the product system models used to perform LCA computations. This article presents a framework for formally disclosing the foreground of an LCA study in a way that permits the computations to be inspected, verified, and reproduced by a reader, provided that the reader has access to the same life cycle inventory and impact characterization resources as the author. The framework can also be used to partition a study into public and private portions, allowing both portions to be critically reviewed but omitting the private information from the disclosure. A disclosure is made up of six components, including three lists of entities in the model and three sparse matrices describing their interconnections. The entity lists make reference to previously-published resources, including background inventory databases and characterized elementary flows, and the disclosure framework requires both author and reader to agree on the meaning of each of these references. The framework contributes to ongoing efforts within and beyond industrial ecology to improve the reproducibility and verifiability of scholarly works, and if implemented, plots a course toward distributed, platform-independent computation and validation of LCA results.

Disclosure of Product System Models in Life Cycle Assessment: Achieving Transparency and Privacy

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

2018-04-23

Submitted after revision to the Journal of Industrial Ecology; April 2018

Abstract

Many of the challenges facing knowledge synthesis from life cycle assessment (LCA) studies stem from the inability of study authors and readers to formally agree on the structure and content of the product system models used to perform LCA computations. This article presents a framework for formally disclosing the foreground of an LCA study in a way that permits the computations to be inspected, verified, and reproduced by a reader, provided that the reader has access to the same life cycle inventory and impact characterization resources as the author. The framework can also be used to partition a study into public and private portions, allowing both portions to be critically reviewed but omitting the private information from the disclosure. A disclosure is made up of six components, including three lists of entities in the model and three sparse matrices describing their interconnections. The entity lists make reference to previously-published resources, including background inventory databases and characterized elementary flows, and the disclosure framework requires both author and reader to agree on the meaning of each of these references. The framework contributes to ongoing efforts within and beyond industrial ecology to improve the reproducibility and verifiability of scholarly works, and if implemented, plots a course toward distributed, platform-independent computation and validation of LCA results.

<Level 1> Introduction

Life cycle assessment (LCA) is a standard methodology to estimate the potential environmental impacts of products or services by modeling the network of industrial processes that must occur to deliver them to a consumer. The technique is well-established, widely practiced, and draws on

an extensive body of standardization and scholarship. Increasingly, the results of LCA studies are appearing in environmental declarations, corporate sustainability reports, and product marketing information. LCA is also gaining prominence as a tool for developing and evaluating environmental policy. At the core of an LCA study is a model of a "product system," which represents a collection of "processes … performing one or more defined functions", that describes the life cycle of a product (ISO 14044, 2006). The product system model, or PSM, encodes information about how a product is manufactured, distributed, used by the consumer, and what happens to it after it is disposed. A product system can be divided into a foreground, which denotes the portions of the life cycle whose operations are directly modeled by the study, and a background, which represents the global industrial system (Clift *et al.*, 1998). While the foreground is often modeled from direct observation or other primary data, background processes are typically drawn from a life cycle inventory (LCI) database prepared by a third party.

Although there has been considerable effort to normalize and harmonize LCI database design methodology and data interchange (UNEP/SETAC, 2011; Recchioni *et al.*, 2013; Mila i Canals *et al.*, 2015; Ingwersen *et al.*, 2015), less attention has been paid to harmonizing PSMs. These models are often highly complex and include countless modeling decisions, approximations, and assumptions made by a study author (Lloyd & Ries, 2007; Reap *et al.*, 2008). Factors such as co-production strategy (Finnveden, 1999; Pelletier *et al.*, 2014) can also render study findings unreliable or highly contingent. Consequently, while comparative evaluations can be made regarding alternative cases or scenarios within a single study, comparisons across studies are much more challenging (Heath & Mann, 2012; Henriksson *et al.*, 2014). Even when relatively simple and homogeneous systems are considered, wide variations can be found in the results of studies from different authors (van der Harst & Potting, 2013; Turconi *et al.*, 2013). A lack of transparency in reporting, particularly regarding definitions of study scope and system boundary, is a crucial challenge to the interpretation of results in comparative analysis (Cleary, 2009; Laurent *et al.*, 2014). A further challenge to transparency can be found in the use of input data that are confidential or proprietary (Kuczenski *et al.*, 2016). One consequence of all these challenges is the high cost

and uneven rigor of critical review, in which the complexities of LCA come face to face with the limitations of current practice (Curran & Young, 2014).

As LCA gains prominence, particularly in the policy realm, these problems become more acute. If LCA is to be used in policy, it will necessarily have the effect of recommending certain product systems, technologies, or approaches at the expense of others; yet in that event it is critically important that there be a consensus among stakeholders that the quantitative results are credible and well-supported (Rainville *et al.*, 2015; McManus *et al.*, 2015). As noted above, this level of consensus is hard to realize, in part because of the conflicting implications inherent in different analytic modes. Plevin et al. famously observed in the strongest terms that careless reporting of attributional LCA study results can distort the significance of findings in a policy context (Plevin *et al.*, 2013), a shortcoming that can be found in other modes as well (Brandão *et al.*, 2014).

An LCA study result is ultimately an assertion that for some constructed PSM, the delivery of a particular reference flow is associated with a certain amount of environmental impact or potential impact. But if the PSM itself is ambiguously stated, then the significance of the result is ambiguous as well. Often the structure and contents of the model are never stated precisely; instead they are *described* in a written report, and the description must be interpreted by a reader. When a study is reviewed, it is often the written report and *not* the PSM itself that is the object of review. ISO 14071 (2014), which provides guidelines on critical review, emphasizes that review of the PSM or individual datasets is optional. Although the general structure for life cycle assessment can be stated formally, there is no current practice for formally describing the PSM and thus, no automatic mechanisms for interpretation or review of published LCA studies. Moreover, there is no current practice that permits a reviewer to verify the correctness of the computations inherent in an LCA study if access to the PSM itself is not available.

Because of the diversity of software and techniques for performing LCA, it is important to develop methods for communicating the structure of the PSM that is not dependent on any particular software or database (Kuczenski *et al.*, 2018). The purpose of this paper is to advance a framework for

the formal description of PSM that would allow for sharing, revision, and critical review of models while also enabling authors to protect the privacy of confidential information. This objective is also in line with the broader imperative throughout industrial ecology to improve transparency and reproducibility of computed results (Hertwich *et al.*, 2018).

<Level 1> Disclosing the Product System Model

<Level 2> Objectives

The task of computing life cycle impact assessment (LCIA) results from a product system model can be described as an "aggregation," in which the data in large, sparse models are combined through linear algebra operations into a small set of numeric scores. Although several equivalent methods for computing aggregated LCI and/or LCIA results have been described in the literature (Suh & Huppes, 2005; Peters, 2007), the computational representation of and exchange of product system models, and the generation of aggregated results, remains software-specific.

Reporting the results of an LCA study implies the existence of two parties with different information: the study author has full knowledge of the product system model; and the reader only has knowledge of what the author has disclosed. Obviously, the level of detail of the disclosure will determine how useful it is to the reader. In this context, the reader will want to know the answers to various review objectives. At the same time the author may have constraints that limit what can be disclosed. The main goal of the framework presented here is to give form to the disclosure so that the requirements of various review objectives, and restrictions associated with various disclosure contraints, can be evaluated clearly.

On the basis of transparency, the framework for disclosure of the PSM should meet the following objectives:

- **Computability**. The result should be computed using only the disclosure, plus resources

available independently to the reader. The computation should follow the consensus understanding of process LCA computation.

- **Completeness / Minimality**. The computation should use all of the information provided in the disclosure.
- **Reproducibility**. The result obtained by the reader should match the result provided by the author.

One key objective of disclosure is to inspect the work of the author. Consequently, the disclosure should be made up of the author's own work product and should exclude, for instance, data that were drawn from a reference database or other independent source and used in an unmodified form. The purpose of this is twofold: to focus the efforts of the reader on the work of the author, and to enable the reader to obtain and inspect the third-party data independently, thus ensuring its integrity.

- **Authority**. The contents of the disclosure should authentically represent the outcomes of the author's labor in constructing the PSM.
- **Originality**. The disclosure should exclude information that is made available by a third party. (It must, however, still include previously-published data that were re-implemented by the author in the study.)

In addition, because of the ubiquity of confidential and sensitive information in LCA, any operational disclosure framework must also meet the following:

- **Privacy**. The disclosure should protect information that is held in confidence by the author.

<Level 2> Mathematical Formulation of LCA

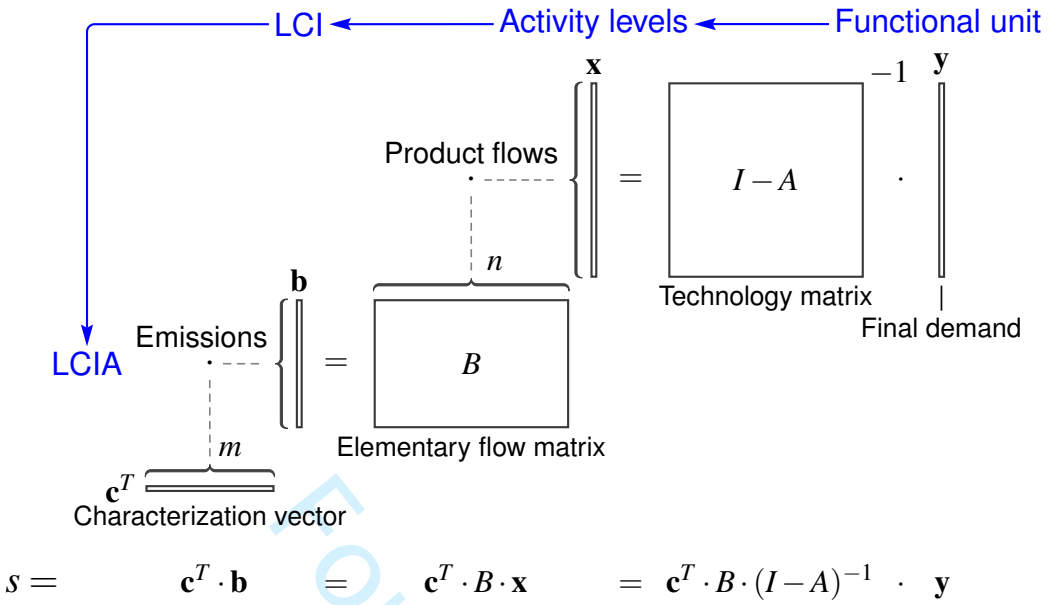
In the most general terms, the LCA problem concerns the interaction of a collection of activities and commodities described by a supply-and-use inventory, in which U is a table of the use of

each commodity by each activity (inputs) and V is a table of the supply of each commodity by each activity (outputs) (Pauliuk *et al.*, 2015). Such an account can always be transformed into a *symmetric* table of inputs and outputs on a commodity-by-commodity or industry-by-industry basis (Eurostat, 2008). Doing so requires the application of various assumptions and transformations to the data, the details of which are not in scope of this paper, but which have recently been reviewed in depth (Majeau-Bettez *et al.*, 2014). Although they originated independently, the "classical" process-based approach and the input-output approach have been shown to be equivalent for the purposes of LCA computation (Suh *et al.*, 2010). Please see the Supporting information for a discussion about this equivalency.

A symmetric input output table can be written as a table of coefficients, in which each column of the matrix provides a "recipe" for creating the corresponding commodity by consuming other commodities (and, if applicable, producing other co-products), equivalent to a "unit process" in ISO LCA, except that the process must have exactly one reference flow, the identity of which is implicit in the column's index. The table of coefficients, commonly known as a *direct requirements matrix*, can be used to compute the output from all activities that is necessary to produce a given commodity. That vector of outputs can then be combined with a second table of coefficients, known as an *emission matrix*, that reports exchanges with the environment associated with the production of each commodity, to determine the cumulative environmental emissions associated with the production of the commodity over its life cycle. Those emissions are individually *characterized* with respect to an environmental impact category of interest, and the characterized emissions summed to report the *impact score* or category indicator, which is the result of the study. The formulation may be summarized:

$$s = \mathbf{c}^T \cdot B \cdot (I - A)^{-1} \cdot \mathbf{y} \tag{1}$$

Here A represents the direct requirements, sometimes known as the *Leontief matrix*; B is the emission matrix; $\mathbf{c}$ is a column vector of characterization factors for the environmental emissions; $\mathbf{y}$ is the externally specified final demand; and s is the numerical impact score or category indicator.



$$s = \mathbf{c}^T \cdot \mathbf{b} = \mathbf{c}^T \cdot B \cdot \mathbf{x} = \mathbf{c}^T \cdot B \cdot (I - A)^{-1} \cdot \mathbf{y}$$

Figure 1: Traditional matrix structure of an LCA computation.

For simplicity, we will consider the computation of a single impact result. However, it is straightforward to imagine the more typical case, in which C is a matrix of characterization vectors, and $\mathbf{s}$ is a vector of results. Eq. 1 is visualized in Figure 1.

The term $(I - A)$ is equivalent to a *technology matrix* in traditional LCA, normalized by column so that each column represents a unit output of a single reference product, and each row represents the use of the corresponding product. The coefficient matrices A (or $I - A$) and B together are called a *life cycle inventory database* (LCIDB), and the functional unit of a study is given by the final demand vector $\mathbf{y}$, which is the product or service whose delivery is to be assessed. Intermediate results of this computation include:

- $\mathbf{x} = (I - A)^{-1} \cdot \mathbf{y}$, the activity level vector;
- $\mathbf{b} = B \cdot \mathbf{x}$, the life cycle inventory;
- $s = \mathbf{c}^T \cdot \mathbf{b}$, the life cycle impact category indicator, the study result.

Some LCA software systems have the capacity to export the A and B matrices for a given study model. However, the contents of commercial LCIDBs are proprietary and subject to licensing

restrictions, and moreover are not usually modified by study authors. Thus exporting A and B does not meet our authority and originality objectives. Additionally, replicating the entire LCIDB is cumbersome because of the large size of the matrices. As a consequence, the matrix formulation in Eq. 1 is unsuitable as a disclosure framework.

<Level 2> Foreground Study

In practice, most LCA studies contain information that is not present in a background database, and LCA software systems generally allow users to augment the standard LCIDBs with their own modeling information, known as the study foreground. To meet our disclosure objectives, it is important to be able to distinguish foreground content from background content in the computation. Background databases are prepared independently of any study and do not typically vary across the studies in which they are used. However, the results of a study with foreground content cannot generally be computed without knowledge of the background.

To understand how a foreground study is constructed on top of a background database, imagine a simple product system that can be fully described by only making reference to processes that exist in a background database. For instance, the use of a computer may be modeled using reference processes for producing and disposing the computer, and a reference process for electricity consumption. Assuming those reference processes are available in a background database, the product system can be represented entirely by specifying a final demand vector $\mathbf{y}$ whose nonzero values correspond to computer production, computer disposal, and electricity. The LCIA results can be computed using Eq. 1.

The same result can be obtained by relocating the $\mathbf{y}$ vector into an augmented A matrix, augmenting the B matrix with a vector of zeros, and replacing the final demand with a vector $\tilde{\mathbf{y}}$ in which the

first entry is 1 and all successive entries are 0:

$$\begin{aligned} s = &\quad \mathbf{c}^T \cdot B \cdot (I - A)^{-1} \cdot \mathbf{y} \\ = &\quad \mathbf{c}^T \cdot \tilde{B} \cdot \left(I - \tilde{A}\right)^{-1} \cdot \tilde{\mathbf{y}} \end{aligned} \tag{2}$$

where $\tilde{A} = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{y} & A \end{bmatrix}$, $\tilde{B} = [\mathbf{0}, B]$, and $\tilde{\mathbf{y}} = [1, 0, 0, \ldots, 0]^T$. It can easily be shown that Eq. 2 holds for any $\mathbf{y}$ (A proof is provided in the electronic supporting information). This is intuitively true because the $\tilde{\mathbf{y}}$ selects the first column of the augmented $\tilde{A}$ matrix as the "final demand," but the "direct requirements" of that column are precisely the final demand from the prior, un-augmented A matrix.

This same workflow can be repeated to build a more elaborate foreground. For instance, a model of private vehicle travel may make use of reference processes for vehicle production and decommissioning, road maintenance, fuel retail, and a model of direct emissions from fuel combustion. This model could occupy an additional column of $\tilde{A}$, with the coefficients for fuel combustion emissions being contained in the corresponding column of $\tilde{B}$. A process for brewing coffee could be created in a third column that requires coffee beans, water, and electricity. Finally, a fourth column could be added that invokes the products from the first three columns in proportion to describe the reference product of traveling to a coffeeshop to work.

In general, a product system model may be constructed through the successive augmentation of a background database with foreground content. As long as the background database is not altered to depend on the newly created foreground, such a study can be written in block triangular form (Kuczenski, 2015):

$$\tilde{A} = \begin{bmatrix} A_f & 0 \\ A_d & A \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} B_f & B \end{bmatrix} \tag{3}$$

Here, the submatrix A_f represents the foreground; A represents the background; the rectangular matrix A_d represents the dependency of the foreground on the background; and the top right submatrix is zero. The ordered $\tilde{B}$ matrix is similarly partitioned into B_f, which includes foreground

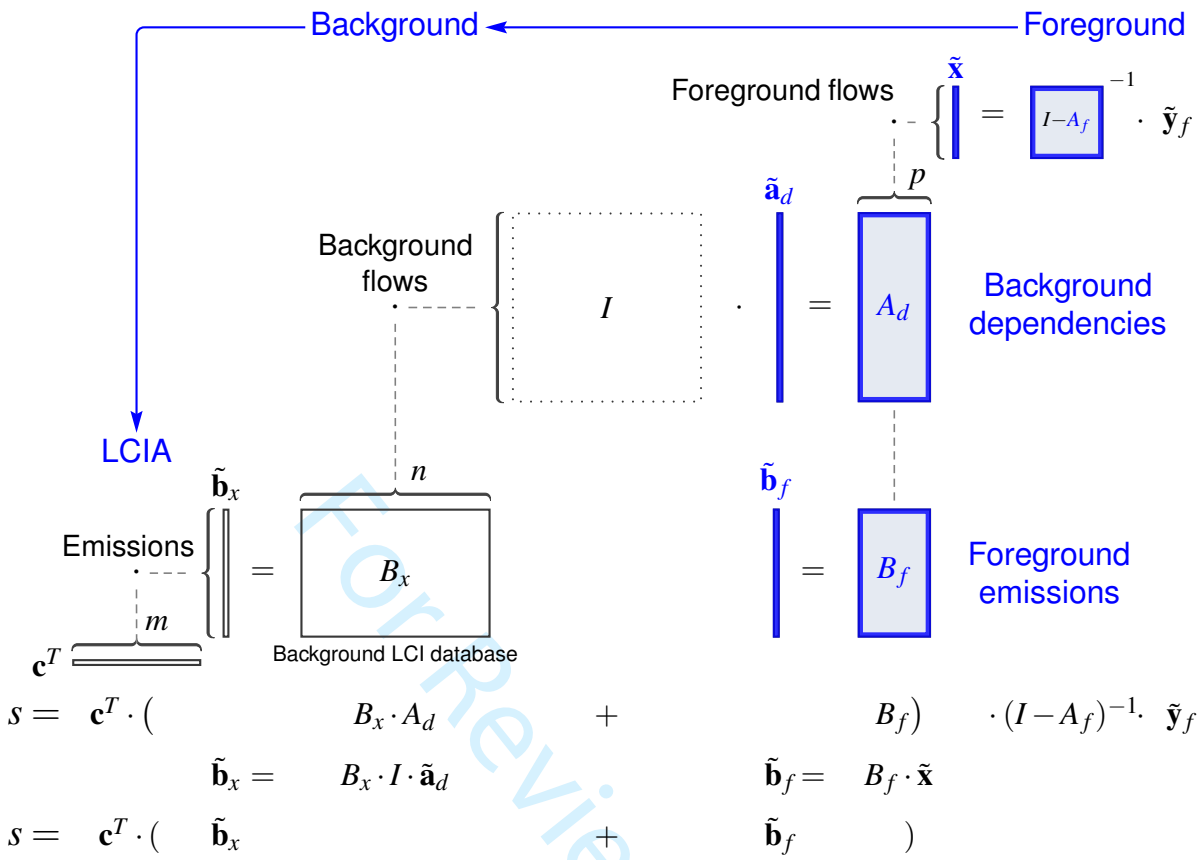


Figure 2: Matrix structure of an LCA foreground study. The background technology matrix has been incorporated into a background LCI database B_x, and replaced in the figure with a placeholder identity matrix I. The foreground and aggregated foreground elements are highlighted.

emissions, and B, which includes background emissions. The augmented $\tilde{A}$ and $\tilde{B}$ together make up an *LCA foreground study*. This formulation can be applied to the vast majority of LCA case studies, and very likely to all case studies prepared using commercial LCA software.[1]

Without loss of generality, it is possible to construct the study such that the *first* column of $\tilde{A}$ delivers the functional unit of the study. We thereby designate the vector $\tilde{\mathbf{y}} = [1, 0, 0, \ldots, 0]^T$ as the *canonical functional unit* of a foreground study, which allows us to introduce convenient notation later on.[2]

It is currently common practice for background database maintainers to pre-compute the life cycle inventory results for their databases. In cases where the background database is used without modification, including the contents of A in $\tilde{A}$ becomes unnecessary. Using the notation $B_x = B \cdot$

$(I-A)^{-1}$ to represent the pre-computed background LCI database, Eq. 1 can be written so that the foreground matrix inversion is performed separately from the computationally costly background computation:

$$s = \mathbf{c}^T \cdot (B_f + B_x \cdot A_d) \cdot (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f \tag{4}$$

where $\tilde{\mathbf{y}}_f$ is a canonical functional unit vector whose length is the same as the rank of A_f. Eq. 4 is derived in the supporting information. Eq. 4 is visualized in Figure 2, with the background technology matrix A subsumed into B_x and replaced by an identity matrix I.

The activity levels of the foreground nodes are represented by $\tilde{\mathbf{x}}$, which is the result of $\tilde{\mathbf{y}}_f$ selecting the first column of $(I-A_f)^{-1}$. This vector can be used to describe an *aggregated foreground* in coefficient form, which delivers the same results as the foreground study given in Eq. 3, but without disclosing any details of the construction of the foreground:

$$\begin{aligned} \tilde{\mathbf{b}}_f &= B_f \cdot \tilde{\mathbf{x}} \\ \tilde{\mathbf{a}}_d &= A_d \cdot \tilde{\mathbf{x}} \\ \tilde{\mathbf{b}}_x &= B_x \cdot \tilde{\mathbf{a}}_d \end{aligned} \tag{5}$$

Here $\tilde{\mathbf{a}}_d$ reports the aggregated foreground's dependency on background processes in background flows per canonical functional unit; $\tilde{\mathbf{b}}_f$ reports foreground emissions per functional unit; and $\tilde{\mathbf{b}}_x$ reports the aggregated background life cycle emissions per functional unit.

The aggregated result can also be computed from the aggregated foreground:

$$\begin{aligned} s &= \mathbf{c}^T \cdot (B_f + B_x \cdot A_d) \cdot \tilde{\mathbf{x}} \\ &= \mathbf{c}^T \cdot (\tilde{\mathbf{b}}_f + B_x \cdot \tilde{\mathbf{a}}_d) \\ &= \mathbf{c}^T \cdot (\tilde{\mathbf{b}}_f + \tilde{\mathbf{b}}_x) \end{aligned} \tag{6}$$

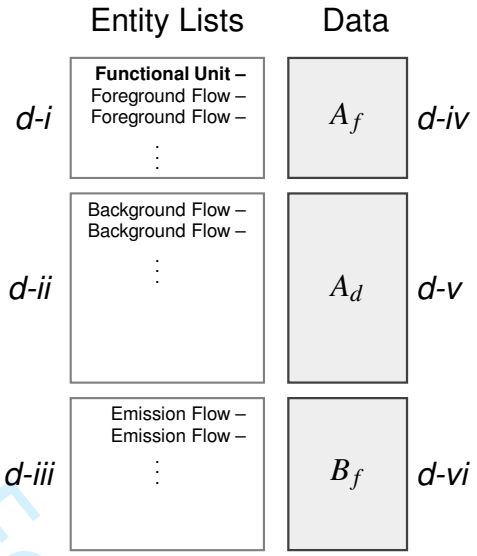


Figure 3: Graphical depiction of the components of a disclosure.

<Level 2> Contents of a Disclosure

The foreground matrix A_f, dependency matrix A_d, and foreground emission matrix B_f now contain all study-specific information. Assuming that the appropriate background LCIDB B_x, and the vector of characterization factors $\mathbf{c}$ (or their product $\mathbf{c} \cdot B_x$) are available and held in common between the author and the reader, then a disclosure that precisely describes A_f, A_d and B_f is sufficient to reproduce the study result, as formulated in Eq. 4. These three submatrices together can be considered a complete representation of the PSM.

Thus, the problem of how to disclose an LCA study design is reduced to the problem of how to communicate the contents of these submatrices accurately. Recognizing that all three matrices are likely to be sparse (that is, that most of the entries are zero), the most efficient disclosure would be to only report non-zero entries. A sparse matrix can be represented by a list of 3-tuples indicating (row, column, value) for each non-zero entry. A formal disclosure of the PSM can be stated in six parts, including three lists of entities and three sets of data values:

d-i. An ordered list of foreground nodes, beginning with the functional unit (rows/columns of A_f);

d-ii. An ordered list of background flows, each making reference to a particular background dataset (rows of A_d, mapping to columns of B_x);

d-iii. An ordered list of exchanges with the environment (rows of B_f, mapping to entries in **c**);

d-iv. A set of 3-tuples for nonzero elements of A_f, in which the row and column are indices into Item *d-i*;

d-v. a set of 3-tuples for nonzero elements of A_d, in which the rows are indices into Item *d-ii* and the columns are indices into Item *d-i*;

d-vi. a set of 3-tuples for nonzero elements of B_f, in which the rows are indices into Item *d-iii* and the columns are indices into Item *d-i*.

Each entity included in items *d-i* through *d-iii* describes a flow of some kind, and its unit of measure must be reported in the disclosure. In both items *d-ii* and *d-iii*, only rows containing non-zero entries need to be mentioned, since the index of each row is common information. The contents of a disclosure are represented graphically in Fig. 3, and two example disclosures drawn from reference databases can be found in the supporting information.

A disclosure containing these six elements satisfies most of the requirements established at the top of this section (the exception is ensuring the privacy of confidential data, which is treated later in the article):

- The result is **computable** using Eq. 4;
- the computation is **complete** and **minimal** because all components are required and no superfluous components are included;
- The results are **reproducible** if the reader is able to accurately interpret the author's references in *d-ii* and *d-iii*;
- The disclosure includes only information for which the author can claim **authority**;
- The disclosure includes all information specific to the study, or for which the study is the

original source, including re-implementations of already-published data.

As mentioned, the proper interpretation of the disclosure requires that the reader accurately interprets the author's references to background databases and elementary flows, including mutually consistent implementation of the LCIA vector **c**. These aspects are out of scope for the purposes of this paper, and we will assume that a shared understanding between author and reader can be reached. However, the challenges of operationalizing this requirement are complex (see the discussion and supporting information).

<Level 1> Reviewing Product System Models

Objectives in critical review vary with the study and the reviewer, but they generally include consistency with the ISO 14044 standard, the validity of methods, appropriateness of data used, correctness of interpretations, and transparency of reporting (ISO 14044, 2006, Section 6.1). Although the object of critical review is typically a written report, this level of review is not sufficient to ensure the correctness of quantitative results. The use of a structured disclosure of a PSM introduces the possibility that reviewers can inspect the model itself and even verify the computations and reported results. This section discusses how a disclosure in the format described above can enable robust review of a product system model. Note that, for purposes of critical review, the objective for data privacy can be omitted because the reviewer may be granted privileged access to confidential data. We return to the data privacy requirement later in this section.

<Level 2> The Foreground System Boundary and Cut-off Flows

The foreground of an LCA study is a collection of nodes that represent points in the inventory model where flows are exchanged. The foreground matrix A_f describes in precise terms what components are included in the product system and how they connect to one another. The foreground is modelled as a weighted directed graph in which there is a 1:1 correspondence between

nodes and product flows. The foreground matrix A_f reports the adjacency and weights of edges in the graph, and thus describes the relationships among these nodes. Some simple foregrounds are described in the supporting information.

Generally, the direct requirements matrix can represent any process-flow model with a 1:1 correspondence between processes and flows. Normally this is understood as a requirement to include only allocated single-output processes. However, the foreground disclosure framework is also well suited to documenting how multi-output processes are transformed into single-output processes through either partitioning allocation or different forms of system expansion. Different approaches for creating foregrounds to model multi-output process are presented in the supporting information.

An implicit "foreground system boundary" can be imagined to contain all the nodes in A_f. Nonzero values in this matrix represent exchanges inside this boundary. If a column in A_f is nonempty, then the corresponding node has exchanges with other nodes in the foreground. On the other hand, if a column in A_f is empty, then that is an indication that the corresponding product flow is crossing the foreground system boundary. In order for this flow to contribute any impacts, it must either be linked to a background process in the dependency matrix, or represented as an emission.

A flow whose entire column consists of 0s throughout A_f, A_d, and B_f is a "cut-off" that exits the model with zero burdens. Specifying cut-off criteria and reporting of cut-off flows is an important part of system boundary definition. Thus, reviewing cut-off flows (including evaluating their significance) is a crucial part of review that is facilitated by having a structured disclosure of the model.

<Level 2> Dependencies and Emissions

The foreground matrix only describes the structure of the PSM; it does not provide any information about inventory or impacts. The inventory requirements and environmental burdens associated with the model are all described in the dependency matrix A_d and the foreground emission

matrix B_f. The dependency matrix shows how foreground nodes depend on activities provided by processes in a background database; the emission matrix shows direct exchanges with the environment. In both cases, the respective columns of A_d and B_f correspond with columns in A_f and denote dependencies and/or emissions of the corresponding nodes. If columns of A_d and/or B_f are empty, then the node is simply a pass-through process that serves to connect different foreground elements. However, if A_d and/or B_f have nonzero entries in that column, then that indicates some transformation process or emission is occurring at that point in the foreground.

Review of the dependencies of foreground nodes must include review of the selection of background processes to which they are connected (i.e. the identities of the columns of B_x referred to in disclosure item *d-ii*, the list of background flows). Likewise, the identities of the emissions (entries in disclosure item *d-iii*) must also be reviewed.

<Level 1> Foreground Aggregation

When an LCA study result is computed, the foreground and background matrices are aggregated into a scalar indicator score (or a small vector of a few scores). The formulation in Eq. 4 and the illustration in Fig. 2 provides a framework for reporting results at varying levels of aggregation. In this section, I present several different aggregation computations that provide equivalent results for an LCA study. After that, I show how a foreground containing private data can be partitioned into public and private portions, each of which can be comprehensively checked by a reviewer having privileged access, and subsequently disclosed in a manner that protects the private data.

<Level 2> Forms of Aggregation

Figure 2 illustrates the LCA computation in various successive aggregation steps, from full matrices to vectors to a scalar value. The level of aggregation provided in a disclosure will constrain

what review objectives can be met by the reader of a study. Different aggregation forms are illustrated in Figure 4.

<Level 3> Single-Output Unit Process

A single-output unit process makes up a single shared column of A_f, A_d, and B_f, as depicted in Figure 4a. Documentation of unit processes is specified in ISO 14048, and several compliant formats exist, the most well-known being the ILCD and Ecospold XML formats. ISO 14048 specifies a distinction between intermediate and elementary flows, but does not include the concept of a model foreground. In a matrix representation, each unit process must have exactly one reference flow, which corresponds to its row and column in A_f. Please see the supporting information for more detail on preparing the A_f matrix.

<Level 3> Complete Foreground

See Figure 4b. A foreground can be viewed as a linked set of unit processes. The collection of processes themselves make up the foreground of the study, and so exchanges among them are recorded in A_f, and exchanges outside the foreground are represented in A_d or B_f. A complete foreground represents the full disclosure of a model, and a reader can replicate and extend it independently. Any modeling technique used to treat a multi-functional process, such as an allocation or system expansion, can only be fully expressed as a foreground with at least as many columns as outputs. For examples of multi-output processes represented as foreground models, please see the supporting information. The example shown in Figure 4b shows five nodes, including one cut-off flow.

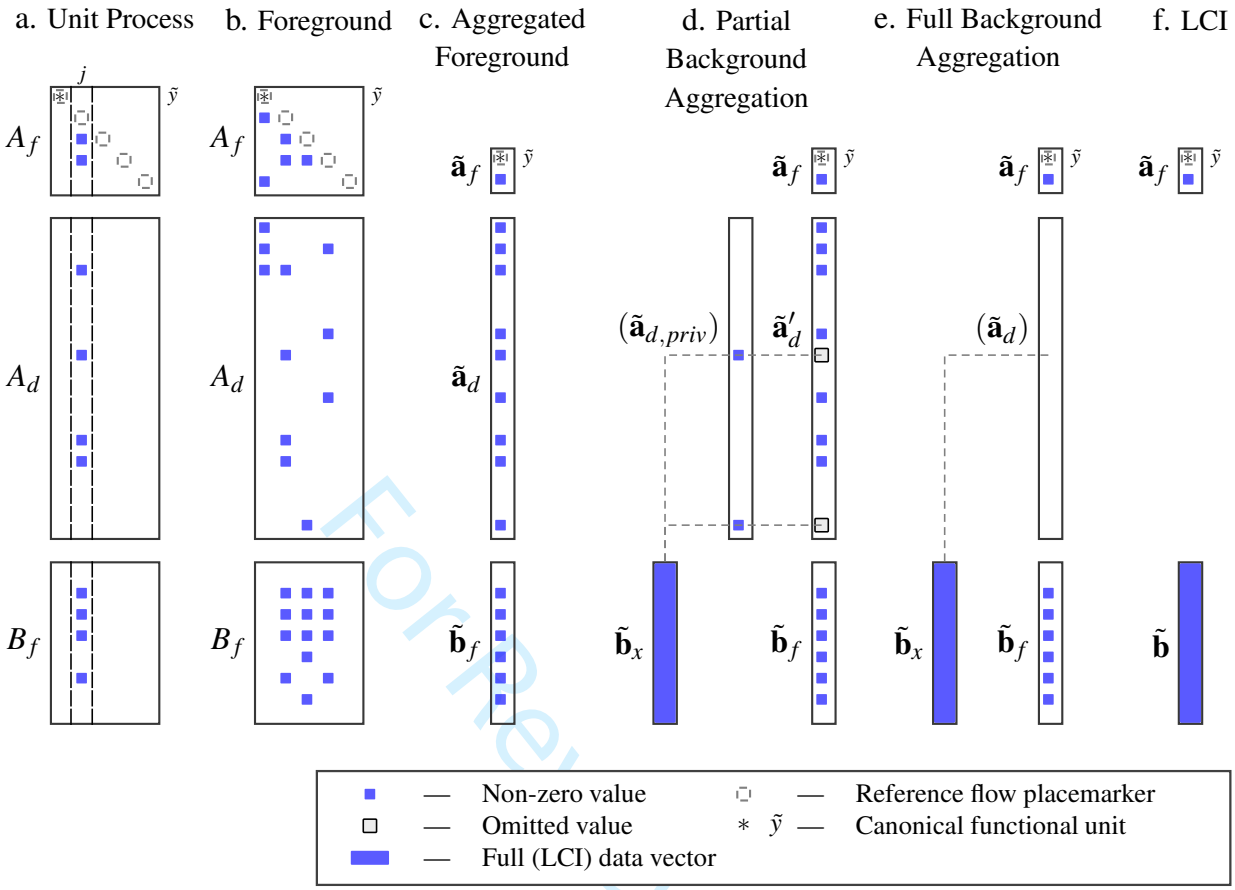


Figure 4: Matrix structures for different forms of foreground aggregation.

<Level 3> Aggregated Foreground

See Figure 4c. Any collection of foreground nodes can also be expressed in aggregated form by performing the computations in Eq. 5. A disclosure of an aggregated foreground resembles the description of a unit process. As in the unit process case, the aggregated foreground has exactly one reference flow, the canonical functional unit $\tilde{\mathbf{y}}$. The aggregated foreground still reports explicit links between the model and a background database and can thus be used to review data set selection.

An aggregated foreground with cut-off flows resembles a multi-output process. After aggregation, foreground flows that correspond to cut-offs must still be reported in a disclosure, while foreground

flows interior to the model can be omitted. This is done with the introduction of a new vector, $\tilde{\mathbf{a}}_f$, which includes the reference flow plus any entries from $\tilde{\mathbf{x}}$ that correspond to cut-off flows.

<Level 3> Partial Background Aggregation

See Figure 4d. Using this approach, the aggregated dependency vector is split into two parts that sum to the original:

$$\tilde{\mathbf{a}}_d = \tilde{\mathbf{a}}_{d,priv} + \tilde{\mathbf{a}}'_d \tag{7}$$

The disclosed dependencies $\tilde{\mathbf{a}}'_d$ are reported, and the private dependencies $\tilde{\mathbf{a}}_{d,priv}$ are replaced with an aggregated background inventory derived from the background database, $\tilde{\mathbf{b}}_{x,agg}$. This approach can be used when a reader lacks access to the background inventory sets referred to in $\tilde{\mathbf{a}}_{d,priv}$, or when the author wishes not to disclose the private dependencies. Foreground emissions $\tilde{\mathbf{b}}_f$ can be reported separately from the aggregation result, but often in current practice they are not distinguished from the background flows.

<Level 3> Full Background Aggregation and LCI

See Figure 4e and f. At this level of aggregation, the entire dependency vector is replaced with an aggregated life cycle inventory vector $\tilde{\mathbf{b}}_x$ derived from the background LCIDB. The reader no longer requires access to any background database to perform the computation, but all dependency information is concealed. As mentioned above, foreground emissions are often not distinguished from the aggregation result; if foreground and background emissions are added together, the disclosure is as shown in Figure 4f. 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 $\mathbf{c}$.

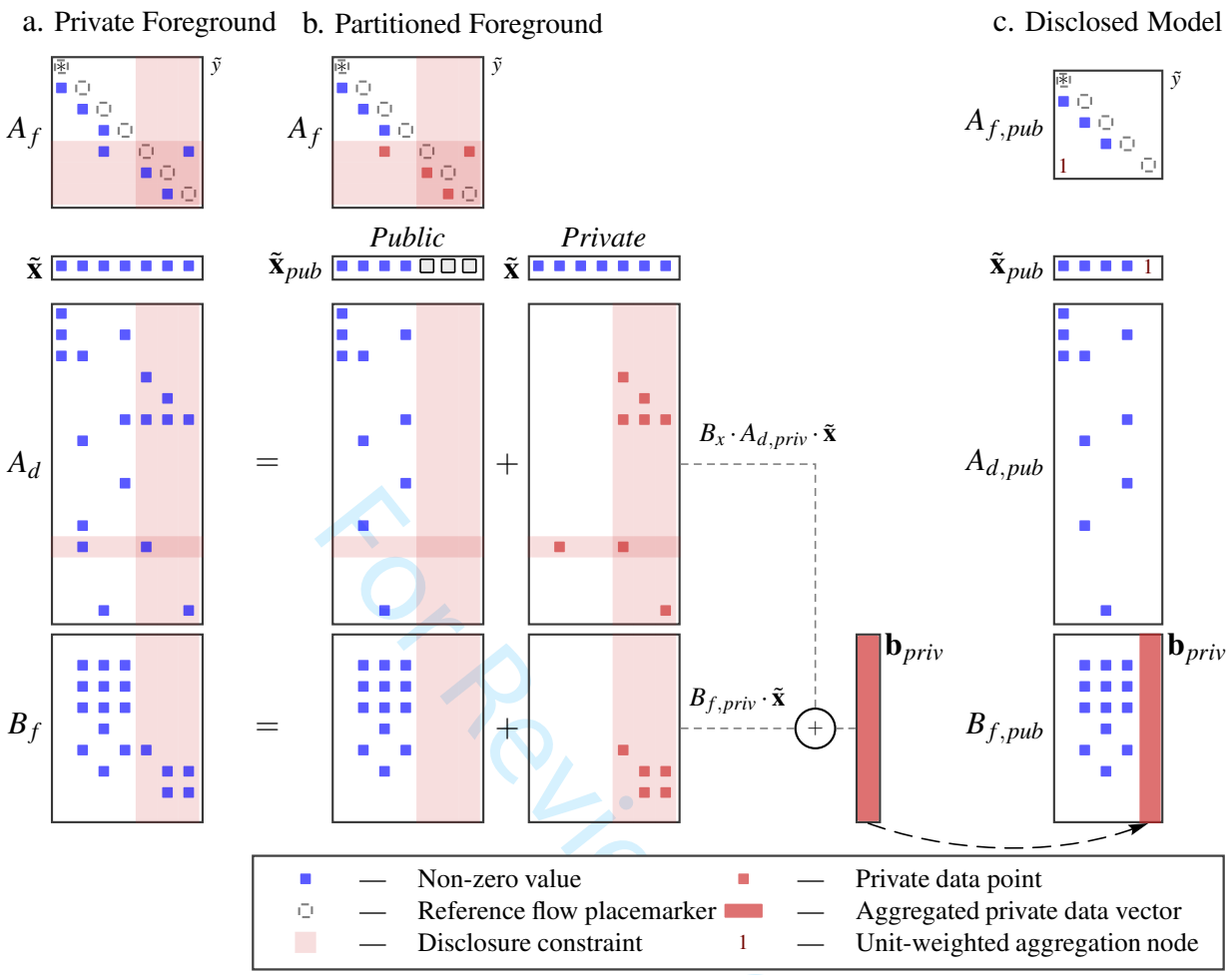


Figure 5: Partitioning the foreground into private and public segments for disclosure. The shaded sections in (a) and (b) indicate disclosure constraints, while in (c) the private data have been replaced with a unit-weighted aggregated column $\mathbf{b}_{priv}$ in B_f.

<Level 2> Reviewable Private Aggregation

Often, a study author may wish to make a partial disclosure of a PSM that retains the privacy of confidential data, while still permitting comprehensive critical review of the complete model. In this case it is necessary to grant a reviewer privileged access to the private data in order to meet the review objectives. The study formulation in Eq. 4 may be used to partition a foreground into public and private portions in order to comply with any applicable constraints on what may be disclosed. The approach is illustrated in Figure 5.

The first step is to identify the constraints on the disclosure. In the framework presented here, constraints may include the locations or values of any subset of nonzero entries in the A_d and B_f matrices, as well as any subset of nodes in A_f. Fig. 5a shows a PSM in which disclosure constraints are indicated by shaded regions of the matrices. In this case the nodes represented by the three rightmost columns of A_f, as well as one entire row of A_d, are to remain secret.

The A_d and B_f matrices are then partitioned into two components that sum to the original:

$$\begin{aligned} A_d &= A_{d,pub} + A_{d,priv} \\ B_f &= B_{f,pub} + B_{f,priv} \end{aligned} \tag{8}$$

This is illustrated in Fig. 5b. Substituted back into Eq. 4, the private portions are aggregated into a private life cycle inventory vector $\mathbf{b}_{priv}$:

$$\mathbf{b}_{priv} = (B_{f,priv} + B_x \cdot A_{d,priv}) \cdot \tilde{\mathbf{x}} \tag{9}$$

while the public portions remain disaggregated:

$$s = \mathbf{c}^T \cdot \left((B_{f,pub} + B_{f,priv}) + B_x \cdot (A_{d,pub} + A_{d,priv}) \right) \cdot \tilde{\mathbf{x}} \tag{10}$$

$$= \mathbf{c}^T \cdot \left((B_{f,pub} + B_x \cdot A_{d,pub}) \cdot \tilde{\mathbf{x}} + \mathbf{b}_{priv} \right) \tag{11}$$

Any foreground nodes that are completely contained within the private partition can be omitted from the disclosed foreground. The aggregation result can then be included as a separate foreground node with a unit weight, and the aggregated inventory vector included as the corresponding column of B_f. This is illustrated in Fig. 5c.

The author may relax the disclosure constraints by reporting the locations of some or all non-zero entries in $A_{d,priv}$ and $B_{f,priv}$ but not disclosing their values, or disclosing a range that includes the actual value. This would enable a reader to understand *that* a certain background process or emission was included in the model (and checked by the critical reviewer) without knowing how

much, which may jointly satisfy review objectives and disclosure constraints in some cases.

<Level 1> Discussion

<Level 2> Comparative Review: Formalizing Model Scope

The disclosure of a PSM contains six parts, including three lists and three tables of sparse matrix data. The graph structure encoded in the foreground matrices can be automatically rendered as a process-flow diagram, supplementing or supplanting the hand-crafted system diagrams that currently accompany ISO reports. While the numeric data are required to critically evaluate and reproduce the study results, substantial information about the study design can be found in the list of background dependencies (*d-ii*) and emissions (*d-iii*). These lists report all the distinct sources of environmental impacts in the model. The cut-off flows, which are foreground nodes that have no nonzero entries in A_f, A_d, or B_f, report on the other hand what aspects were explicitly excluded from the model.

Together, these lists can be considered a functional definition of the study scope that can be used to conduct a qualitative comparison of multiple studies by different authors. Background database choice, dataset selection, and version information can all be made available in disclosure item *d-ii*, thereby streamlining what is presently an interpretive and study-specific process of extracting this information from written text. Similarly, the set of elementary flows explicitly modeled in the study foreground (item *d-iii*) expose whether any study-specific modeling of direct emissions was performed and what flows were included.

Moreover, because the disclosure of the PSM includes only references to background data without including the data themselves, it is trivial for item *d-ii* to include references to a variety of background databases without adding any additional requirements. For instance, a study that makes reference to a set of EPDs published by one firm, while using a background LCI database from

another firm, can express both dependencies in an analogous way, even if the two sources are published independently.

<Level 2> Privacy and Partial Aggregation

Partial aggregation is already used in practice to conceal private data, but current methods do not provide a way for a reviewer to validate the result. If a study containing confidential information is split into public and private partitions as in Eq. 11 and Fig. 5, then the results of the computation can be validated directly by the reviewer. To achieve this, the reviewer must be able to confirm that the partitioning is valid according to Eq. 8, and that $\mathbf{b}_{priv}$ was computed correctly. This requires that the reviewer be granted access to both the public and private partitions. However, the private information used in the computation can still be withheld from the disclosure.

A further advantage to this approach is that the reader can evaluate easily what fraction of the overall impact score is accounted for by the public versus the private portions of the model. A "disclosure completeness" metric φ can be defined as the fraction of the overall score which is accounted for by the disclosed portion of the PSM:

$$\varphi = \frac{\mathbf{c}^T \cdot (\mathbf{b} - \mathbf{b}_{priv})}{\mathbf{c}^T \cdot \mathbf{b}} = 1 - \frac{\mathbf{c}^T \cdot \mathbf{b}_{priv}}{\mathbf{c}^T \cdot \mathbf{b}} \tag{12}$$

where $\mathbf{b}$ is the complete life cycle inventory. The value of this metric can provide an indication of the level of transparency of the disclosure with respect to a given impact category indicator.

<Level 2> Attacks on Privacy

It is not thought possible for a reader to discern any information about a disaggregated system solely from an aggregated disclosure such as $\mathbf{b}_{priv}$ (UNEP/SETAC, 2011, Ch. 3), although that assumption has never been rigorously tested. Compressed sensing is a signal processing technique

that seeks solutions to an underdetermined linear system, in which there are more unknowns than equations, based on the assumption that the solution is sparse (Donoho, 2006). A form of linear optimization called "basis pursuit" can be used to try to detect the signal.

In the LCA case, if the private LCI result $\mathbf{b}_{priv}$ includes inventory data from a background database but no foreground emissions, then it can be written as:

$$\mathbf{b}_{priv} = B_x \cdot \mathbf{a}_{d,priv} \tag{13}$$

Typically this equation is highly underdetermined (In ecoinvent, for instance, B_x has roughly 1,800 rows and over 12,000 columns). If the private input $\mathbf{a}_{d,priv}$ contains only a few nonzero elements, and the identity of B_x is known, then $\mathbf{a}_{d,priv}$ may be vulnerable to a basis pursuit attack. Further research is required to evaluate whether this attack can be implemented in a real situation, and how it may be defended against.

<Level 2> Stable Semantic References to Study Elements

In practice, while it is easy to reproduce a set of sparse matrices, there is considerable potential for ambiguity in identifying the background datasets and emissions to which those matrix entries correspond. In order to achieve the transparency and reproducibility promised by this disclosure framework, it must be possible for study authors and readers to agree on the meaning of the references contained in items *d-ii* and *d-iii*. The FAIR guiding principles, which were designed to help scientists organize data for Findability, Accessibility, Interoperability, and Reusability (Wilkinson *et al.*, 2016), provide valuable perspective on this goal.

Central to many of the FAIR guidelines is the use of linked semantic data (Bizer *et al.*, 2009), in which an object to be interpreted is signified by a link to a resource on the World Wide Web, typically referred to as a Uniform Resource Identifier (URI) or a hyperlink. This link serves as a unique identifier and a resolvable reference to metadata, and the linked content can be curated by

the study authors and other members of the community (Khan *et al.*, 2011).

Before the goal of easily reproducible foreground models can be realized, this linked data foundation must be laid. Background database providers should ensure that their users have access to *stable semantic references* for publicly available datasets that denote a particular activity in a particular database version and configuration. These references should take the form of URIs that can be accessed using any Internet browser and can provide both author and reader with documentary information describing the referred dataset, including version information; database configuration; the available reference flows and their dimensions (quantitative units of measure); and a mechanism for the reader to obtain access to the exchange data or LCI/LCIA computations for the purposes of model validation. A dependency reference in disclosure item *d-ii* should include both an activity and a reference flow.

Similarly, elementary flows and contexts should be given stable references that are held in commomn across data providers. Current research has revealed widespread challenges to finding consistency on the identities of elementary flows despite the community's longstanding awareness of the problem (Speck *et al.*, 2015; Herrmann & Moltesen, 2015), and several mutually inconsistent reference flow sets now exist (Edelen *et al.*, 2017). The conventional understanding of a "flow" as comprising a substance and a context together (e.g. methane, emissions to air) causes a combinatorial increase in the number of flows in a database, and also multiplies the potential points of disagreement across sources. I reiterate Edelen et al.'s (ibid.) suggestion to explicitly regard substances (which can easily include non-material "flowables" such as land occupation and transformation) and contexts as independent semantic classes. In this case, an elementary flow entry in the disclosure item *d-iii* would require both a flowable and a context.

<Level 2> Recommendations for Advancing LCA Practice

The most transformative aspect of the disclosure framework is its ability to describe the productive output of LCA modeling in a way that transcends differences in software environments and

databases. A product system model described in the disclosure framework can be implemented in any software system, though the manner of doing so, as well as availability of advanced features for analysis and visualization, varies widely.

To advance LCA practice, the community should embrace the notion of a software-independent PSM description and consider how to best achieve it. The SETAC North America roadmap module for PSM description and revision (Kuczenski *et al.*, 2018) includes milestones toward this goal. The disclosure framework presented here is offered as a candidate for Milestone 2.1, "a minimal description of a PSM." The community should give this proposal a robust critical evaluation, and researchers should consider how to conceptualize their own models in terms of the framework. Other milestones and cross-cutting issues are of crucial interest:

(a) LCI database providers must provide their users with unambiguous, static, resolvable references to background datasets (Milestone 1.1) so that their use in a model may be documented precisely (Milestone 3.1);

(b) The transformation of multi-functional processes to single-function processes should be described using the framework (Milestones 1.2 and 1.4; see supporting information);

(c) Software makers should enable their users to automatically generate lists of foreground, background, and emission flows a model foreground (disclosure items *d-i* through *d-iii*) and to export foreground matrices (disclosure items *d-iv* through *d-vi*) from a model in a concise format (Milestones 2.2 and 3.4);

(d) During ISO critical review, practitioners and reviewers should strive to come to agreement on the formal structure and contents of the model being reviewed;

(e) The LCA research community, including interested parties within the International Society for Industrial Ecology, the UNEP/SETAC Life Cycle Initiative, and other organizations, should collectively pursue consistent, shared definitions of quantities of measure, flows, and contexts used in LCIA. LCIA method developers should immediately begin using these con-

sensus definitions instead of their own internal lists when publishing flow characterization data, so that software makers can interpret the publications correctly.

As these milestones are pursued and achieved, it will be possible for a standalone description of a PSM to be computed and extended by another party, regardless of the software used to construct the model originally. This will tremendously improve the capacity for critical review and reuse of LCA models.

<Level 1> Conclusion

The current practice of documenting a PSM with a written report sharply limits the ability of readers to interpret and reuse the data. It also results in the model being frozen to a particular configuration or set of configurations used to generate the published results. Any review objectives involving sensitivity of the model to parameter variations and/or alternative scenarios are limited to the selections considered by the author. When a study result is published in an aggregated form, often it is beyond the capacity of the critical reviewer to validate the aggregation result, even if the reviewer has privileged access to confidential materials used in preparing the report.

This paper presents a possible solution by providing a mathematical formalization of the foreground of an LCA study and a functional specification for disclosing the PSM. The non-quantitative portions of the disclosure make up a formal definition of study scope and the boundary of the foreground system. The disclosure framework also allows modelers to express precisely how multifunctionality is handled through allocation or system expansion. A data user or critical reviewer with access to the six disclosure elements, and also adequate access to background datasets and elementary flows used in the model, can reproduce the LCA computation in Eq. 4, and ensure the reported result is correct. Once equipped with the PSM, the reviewer or data user can potentially go much further. Practitioners who are able to reproduce the PSM from the disclosure can also modify it, adding or removing elements, altering dataset selections, applying other impact assess-

ment methods, and considering alternative allocation methods. Portions of the model which have been pre-aggregated can remain private in the disclosure. Further research is required to determine whether or under what conditions aggregation results can be reverse-engineered.

If data providers also make available stable semantic references to background data sets and elementary flow characterizations, then it becomes possible for a stand-alone description of a PSM to be used to reproduce an LCA result. The same description can be extended or incorporated into subsequent studies by other authors. From this prescription, a framework for distributed, platform-independent LCA computation can be imagined.

Acknowledgments

Many thanks to the anonymous reviewers for significant improvement of the manuscript. This work was partially supported by the US National Science Foundation (CCF-1442966).

Notes

[1]It does not apply to cases in which the background database must be altered to adjust for double counting, such as in so-called "integrated hybrid" models in which all quadrants contain non-zero cells (Suh *et al.*, 2004). These models are more complex than the models typically built by LCA practitioners, and require more sophisticated computation. In conventional LCA practice it is inadvisable to alter background databases to adapt them to study conditions. Instead, background processes can be replicated in the foreground, where they can be adapted as needed (cf. Bourgault *et al.* (2012)), without affecting the integrity of the background database.

[2]Of course, the "functional unit" does not need to be limited to a unit of any given output—instead, it *defines* a unit of output. For instance, if the functional unit of a study is 1,000 loaves of bread, and one loaf of bread weighs 0.454 kg, and the unit output from column 567 of the A matrix is 1 kg of bread, then the canonical functional unit would be realized by entering 454 in column 1, row 567, leaving the rest of that column zero.

References

Bizer, Christian, Heath, Tom, & Berners-Lee, Tim. 2009. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, **5**(3), 1–22. doi:10.4018/jswis.2009081901.

Bourgault, Guillaume, Lesage, Pascal, & Samson, Réjean. 2012. Systematic disaggregation: a hybrid LCI computation algorithm enhancing interpretation phase in LCA. *Int. J. Life Cycle Assess.*, **17**(6), 774–786. doi:10.1007/s11367-012-0418-7.

Brandão, Miguel, Clift, Roland, Cowie, Annette, & Greenhalgh, Suzie. 2014. The Use of Life Cycle Assessment in the Support of Robust (Climate) Policy Making: Comment on "Using Attributional Life Cycle Assessment to Estimate Climate-Change Mitigation …". *Journal of Industrial Ecology*, **18**(3), 461–463. doi:10.1111/jiec.12152.

Cleary, Julian. 2009. Life cycle assessments of municipal solid waste management systems: a comparative analysis of selected peer-reviewed literature. *Environment international*, **35**(8), 1256–1266. PMID: 19682746. doi:10.1016/j.envint.2009.07.009.

Clift, R., Frischknecht, R., Huppes, G., Tillman, A.-M., & Weidema, B. 1998. *Toward a Coherent Approach to Life Cycle Inventory Analysis. Report of the Working Group on Inventory Enhancement*. Tech. rept. SETAC-Europe.

Curran, Mary Ann, & Young, Steven B. 2014. Critical review: a summary of the current state-of-practice. *Int. J. Life Cycle Assess.*, **19**(10), 1667–1673. doi:10.1007/s11367-014-0778-2.

Donoho, D.L. 2006. Compressed sensing. *IEEE Transactions on Information Theory*, **52**(4), 1289–1306. doi:10.1109/tit.2006.871582.

Edelen, Ashley, Ingwersen, Wesley W., Rodríguez, Cristina, Alvarenga, Rodrigo A. F., de Almeida, Artur Ribeiro, & Wernet, Gregor. 2017. Critical review of elementary flows in LCA data. *The International Journal of Life Cycle Assessment*, jul. doi:10.1007/s11367-017-1354-3.

Eurostat. 2008. *Eurostat Manual of Supply, Use and Input-Output Tables*. Tech. rept. KS-RA-07-013-EN. Eurostat, Luxembourg.

Finnveden, Göran. 1999. Methodological aspects of life cycle assessment of integrated solid waste management systems. *Resources, Conservation and Recycling*, **26**(3-4), 173–187. doi:10.1016/s0921-3449(99)00005-1.

Heath, Garvin A., & Mann, Margaret K. 2012. Background and Reflections on the Life Cycle Assessment Harmonization Project. *Journal of Industrial Ecology*, **16**, S8–S11. doi:10.1111/j.1530-9290.2012.00478.x.

Henriksson, Patrik John Gustav, Guinée, Jeroen Bartholomeus, Heijungs, Reinout, Koning, Arjan, & Green, Darren Michael. 2014. A protocol for horizontal averaging of unit process data-including estimates for uncertainty. *The International Journal of Life Cycle Assessment*, **19**(2), 429–436. doi:10.1007/s11367-013-0647-4.

Herrmann, Ivan T., & Moltesen, Andreas. 2015. Does it matter which Life Cycle Assessment (LCA) tool you choose? – a comparative assessment of SimaPro and GaBi. *Journal of Cleaner Production*, **86**(jan), 163–169. doi:10.1016/j.jclepro.2014.08.004.

Hertwich, Edgar, Heeren, Niko, Kuczenski, Brandon, Majeau-Bettez, Guillaume, Myers, Rupert J., Pauliuk, Stefan, Stadler, Konstantin, & Lifset, Reid. 2018. Nullius in Verba: Advancing Data Transparency in Industrial Ecology. *Journal of Industrial Ecology*, **22**(1), 6–17. doi:10.1111/jiec.12738.

Ingwersen, Wesley W., Hawkins, Troy R., Transue, Thomas R., Meyer, David E., Moore, Gary, Kahn, Ezra, Arbuckle, Peter, Paulsen, Heidi, & Norris, Gregory A. 2015. A new data architecture for advancing life cycle assessment. *Int. J. Life Cycle Assess.*, **20**(4), 520–526. doi:10.1007/s11367-015-0850-6.

ISO 14044. 2006. *Environmental management — Life cycle assessment — Requirements and guidelines*. ISO, Geneva, Switzerland.

ISO 14071. 2014. *Environmental management — Life cycle assessment — Critical review processes and reviewer competencies: Additional requirements and guidelines to ISO 14044:2006*. ISO, Geneva, Switzerland.

Khan, Huda, Caruso, Brian, Corson-Rikert, Jon, Dietrich, Dianne, Lowe, Brian, & Steinhart, Gail. 2011. DataStaR: Using the Semantic Web approach for Data Curation. *International Journal of Digital Curation*, **6**(2), 209–221. doi:10.2218/ijdc.v6i2.197.

Kuczenski, Brandon. 2015. Partial ordering of life cycle inventory databases. *The International Journal of Life Cycle Assessment*, **20**(12), 1673–1683. doi:10.1007/s11367-015-0972-x.

Kuczenski, Brandon, Sahin, Cetin, & Abbadi, Amr El. 2016. Privacy-preserving aggregation in life cycle assessment. *Environment Systems and Decisions*, **37**(1), 13–21. doi:10.1007/s10669-016-9620-7.

Kuczenski, Brandon, Marvuglia, Antonino, Astudillo, Miguel F., Ingwersen, Wesley W., Satterfield, M. Barclay, Evers, David P., Koffler, Christoph, Navarrete, Tomás, Amor, Ben, & Laurin, Lise. 2018. LCA capability roadmap—product system model description and revision. *The International Journal of Life Cycle Assessment*, feb. doi:10.1007/s11367-018-1446-8.

Laurent, Alexis, Clavreul, Julie, Bernstad, Anna, Bakas, Ioannis, Niero, Monia, Gentil, Emmanuel, Christensen, Thomas H., & Hauschild, Michael Z. 2014. Review of LCA studies of solid waste management systems – Part II: Methodological guidance for a better practice. *Waste Management*, **34**(3), 589–606. doi:10.1016/j.wasman.2013.12.004.

Lloyd, Shannon M., & Ries, Robert. 2007. Characterizing, Propagating, and Analyzing Uncertainty in Life-Cycle Assessment: A Survey of Quantitative Approaches. *Journal of Industrial Ecology*, **11**(1), 161–179. doi:10.1162/jiec.2007.1136.

Majeau-Bettez, Guillaume, Wood, Richard, & Strømman, Anders Hammer. 2014. Unified Theory of Allocations and Constructs in Life Cycle Assessment and Input-Output Analysis. *Journal of Industrial Ecology*, **18**(5), 747–770. doi:10.1111/jiec.12142.

McManus, Marcelle C., Taylor, Caroline M., Mohr, Alison, Whittaker, Carly, Scown, Corinne D., Borrion, Aiduan Li, Glithero, Neryssa J., & Yin, Yao. 2015. Challenge clusters facing LCA in environmental decision-making—what we can learn from biofuels. *The International Journal of Life Cycle Assessment*, **20**(10), 1399–1414. doi:10.1007/s11367-015-0930-7.

Mila i Canals, L., Asselin, A.-C., Cruypenninck, H., Liu, S., Ingwersen, W., & Macdonald, T. 2015. Towards a Global Network of Interoperable LCA Databases. *In: Proceedings of SETAC Europe 25th Annual Meeting (SETAC Europe 2015)*.

Pauliuk, Stefan, Majeau-Bettez, Guillaume, & Müller, Daniel B. 2015. A General System Structure and Accounting Framework for Socioeconomic Metabolism. *Journal of Industrial Ecology*, **19**(5), 728–741. doi:10.1111/jiec.12306.

Pelletier, Nathan, Ardente, Fulvio, Brandão, Miguel, Camillis, Camillo De, & Pennington, David. 2014. Rationales for and limitations of preferred solutions for multi-functionality problems in LCA: is increased consistency possible? *The International Journal of Life Cycle Assessment*, **20**(1), 74–86. doi:10.1007/s11367-014-0812-4.

Peters, Glen P. 2007. Efficient algorithms for Life Cycle Assessment, Input-Output Analysis, and Monte-Carlo Analysis. *Int. J. Life Cycle Assess.*, **12**(6), 373–380. doi:10.1065/lca2006.06.254.

Plevin, Richard J., Delucchi, Mark A., & Creutzig, Felix. 2013. Using Attributional Life Cycle Assessment to Estimate Climate-Change Mitigation Benefits Misleads Policy Makers. *Journal of Industrial Ecology*, **18**(1), 73–83. doi:10.1111/jiec.12074.

Rainville, Anne, Hawkins, Richard, & Bergerson, Joule. 2015. Building consensus in life cycle assessment: the potential for a Canadian product category rules standard to enhance credibility in greenhouse gas emissions estimates for Alberta's oil sands. *Journal of Cleaner Production*, **103**(sep), 525–533. doi:10.1016/j.jclepro.2014.06.067.

Reap, John, Roman, Felipe, Duncan, Scott, & Bras, Bert. 2008. A survey of unresolved problems in life cycle assessment Part 1: goal and scope and inventory analysis. *Int. J. Life Cycle Assess.*, **13**(4), 290–300.

Recchioni, Marco, Mathieux, Fabrice, Goralczyk, Malgorzata, & Schau, Erwin M. 2013. *ILCD Data Network and ELCD Database: current use and further needs for supporting Environmental Footprint and Life Cycle Indicator Projects*. Tech. rept. EUR 25744 EN. European Commission, Joint Research Centre, Luxembourg.

Speck, Ricky, Selke, Susan, Auras, Rafael, & Fitzsimmons, James. 2015. Life Cycle Assessment Software: Selection Can Impact Results. *Journal of Industrial Ecology*, **20**(1), 18–28. doi:10.1111/jiec.12245.

Suh, S., Lenzen, M., Treloar, G.J., Hondo, H., Horvath, A., Huppes, G., Jolliet, O., Klann, U., Krewitt, W., Moriguchi, Y., Munksgaard, J., & Norris, G. 2004. System boundary selection in life-cycle inventories using hybrid approaches. *Environ. Sci. Technol.*, **38**(3), 657–664. DOI: 10.1021/es0263745.

Suh, Sangwon, & Huppes, Gjalt. 2005. Methods for Life Cycle Inventory of a product. *Journal of Cleaner Production*, **13**(7), 687 – 697. doi:http://dx.doi.org/10.1016/j.jclepro.2003.04.001.

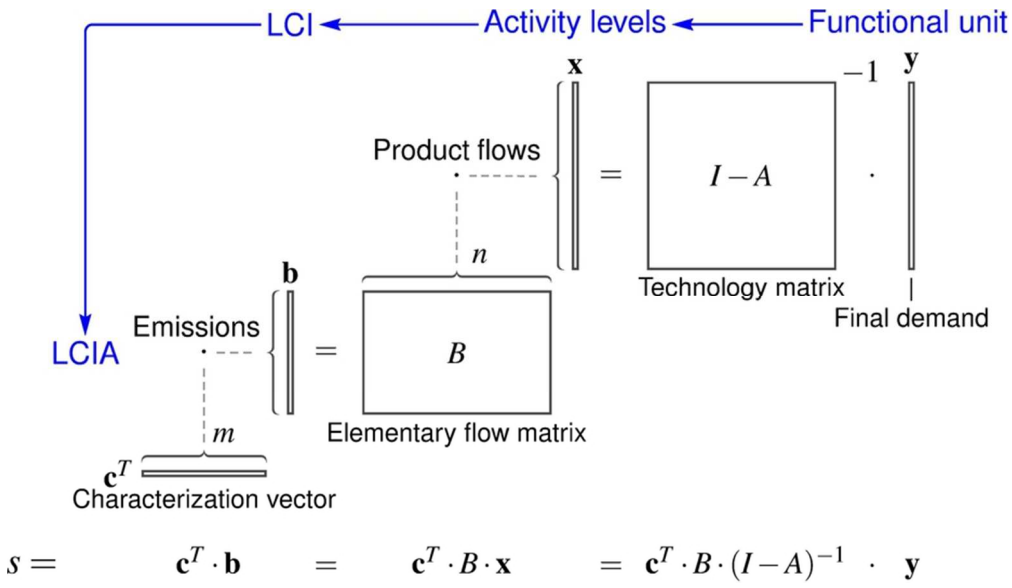
Suh, Sangwon, Weidema, Bo, Schmidt, Jannick Hoejrup, & Heijungs, Reinout. 2010. Generalized Make and Use Framework for Allocation in Life Cycle Assessment. *J. Ind. Ecol.*, **14**(2), 335–353. doi:10.1111/j.1530-9290.2010.00235.x.

Turconi, Roberto, Boldrin, Alessio, & Astrup, Thomas. 2013. Life cycle assessment (LCA) of electricity generation technologies: Overview, comparability and limitations. *Renewable and Sustainable Energy Reviews*, **28**(dec), 555–565. doi:10.1016/j.rser.2013.08.013.

UNEP/SETAC. 2011. *Global Guidance Principles for Life Cycle Assessment Databases*. Tech. rept. United Nations Environment Programme.

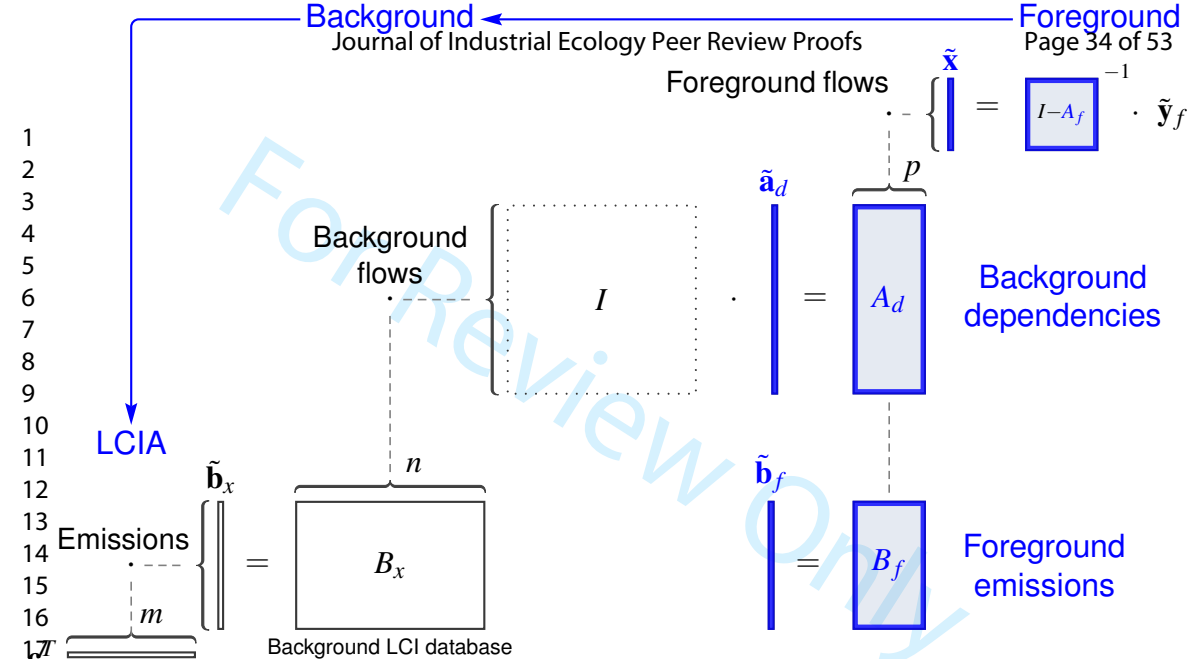
van der Harst, Eugenie, & Potting, José. 2013. A critical comparison of ten disposable cup LCAs. *Environmental Impact Assessment Review*, **43**(nov), 86–96. doi:10.1016/j.eiar.2013.06.006.

Wilkinson, Mark D., Dumontier, Michel, Aalbersberg, IJsbrand Jan, Appleton, Gabrielle, Axton, Myles, Baak, Arie, Blomberg, Niklas, Boiten, Jan-Willem, da Silva Santos, Luiz Bonino, Bourne, Philip E., Bouwman, Jildau, Brookes, Anthony J., Clark, Tim, Crosas, Mercè, Dillo, Ingrid, Dumon, Olivier, Edmunds, Scott, Evelo, Chris T., Finkers, Richard, Gonzalez-Beltran, Alejandra, Gray, Alasdair J.G., Groth, Paul, Goble, Carole, Grethe, Jeffrey S., Heringa, Jaap, 't Hoen, Peter A.C, Hooft, Rob, Kuhn, Tobias, Kok, Ruben, Kok, Joost, Lusher, Scott J., Martone, Maryann E., Mons, Albert, Packer, Abel L., Persson, Bengt, Rocca-Serra, Philippe, Roos, Marco, van Schaik, Rene, Sansone, Susanna-Assunta, Schultes, Erik, Sengstag, Thierry, Slater, Ted, Strawn, George, Swertz, Morris A., Thompson, Mark, van der Lei, Johan, van Mulligen, Erik, Velterop, Jan, Waagmeester, Andra, Wittenburg, Peter, Wolstencroft, Katherine, Zhao, Jun, & Mons, Barend. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, **3**(mar), 160018. doi:10.1038/sdata.2016.18.



Traditional matrix structure of an LCA computation.

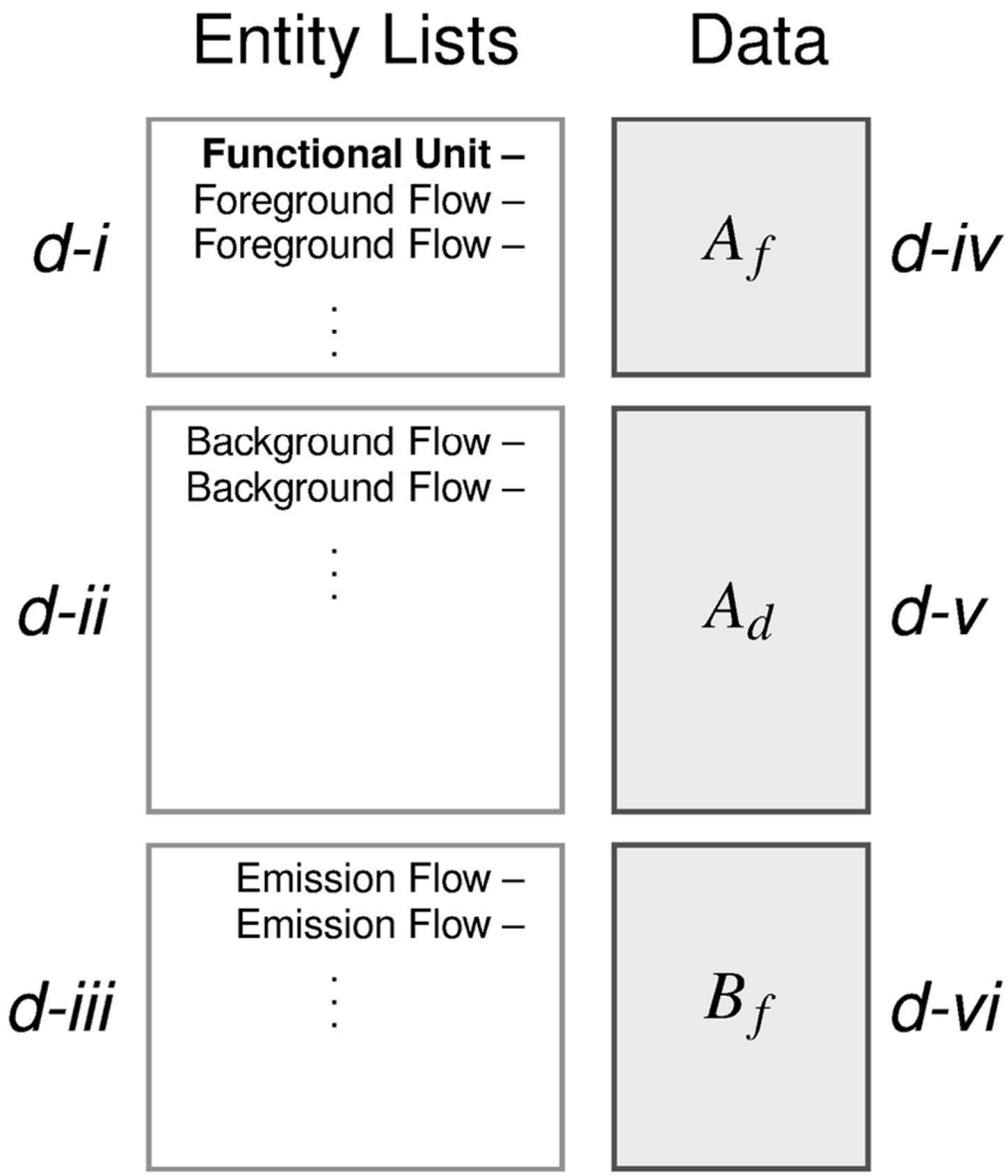
78x44mm (300 x 300 DPI)



$$\mathbf{c}^T \cdot \left(B_x \cdot A_d + B_f \right) \cdot (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f$$

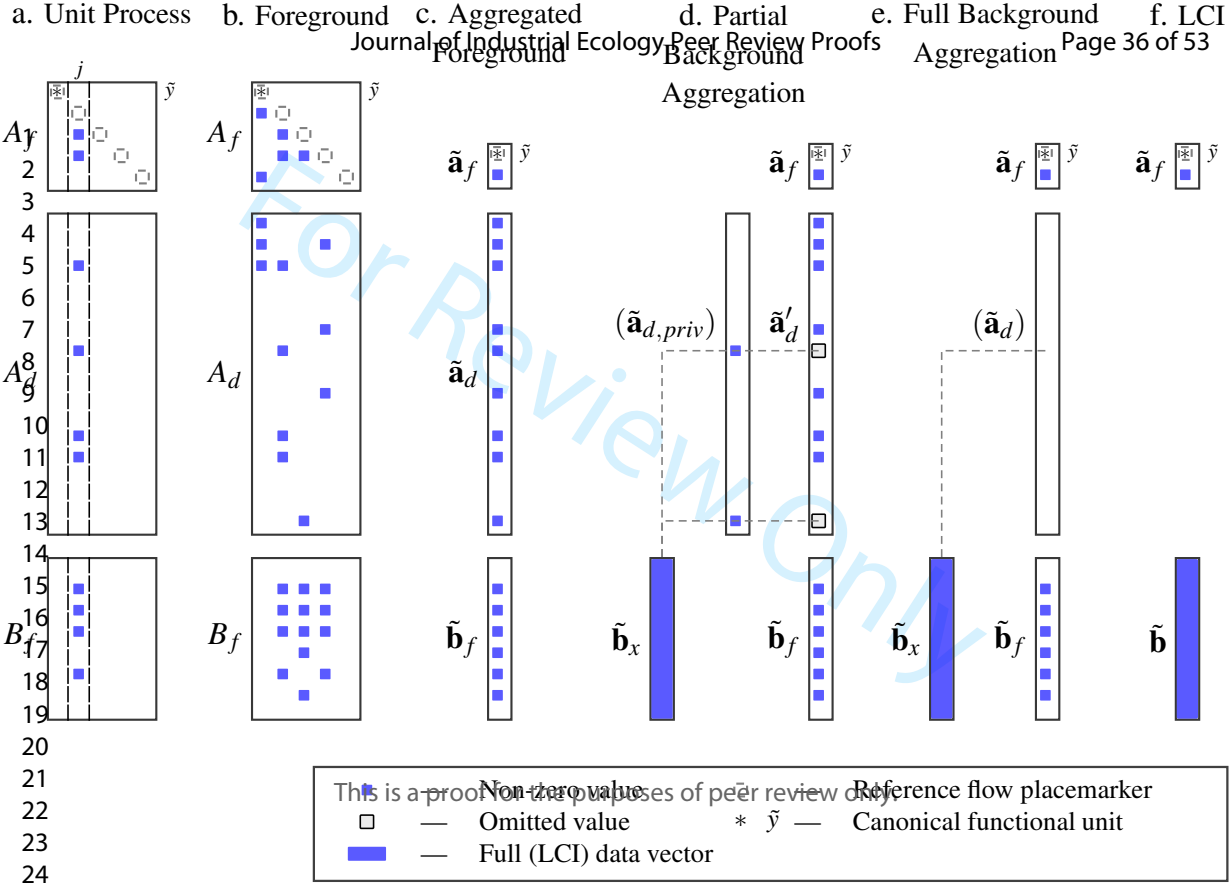
$$\tilde{\mathbf{b}}_x = B_x \cdot I \cdot \tilde{\mathbf{a}}_d \qquad \tilde{\mathbf{b}}_f = B_f \cdot \tilde{\mathbf{x}}$$

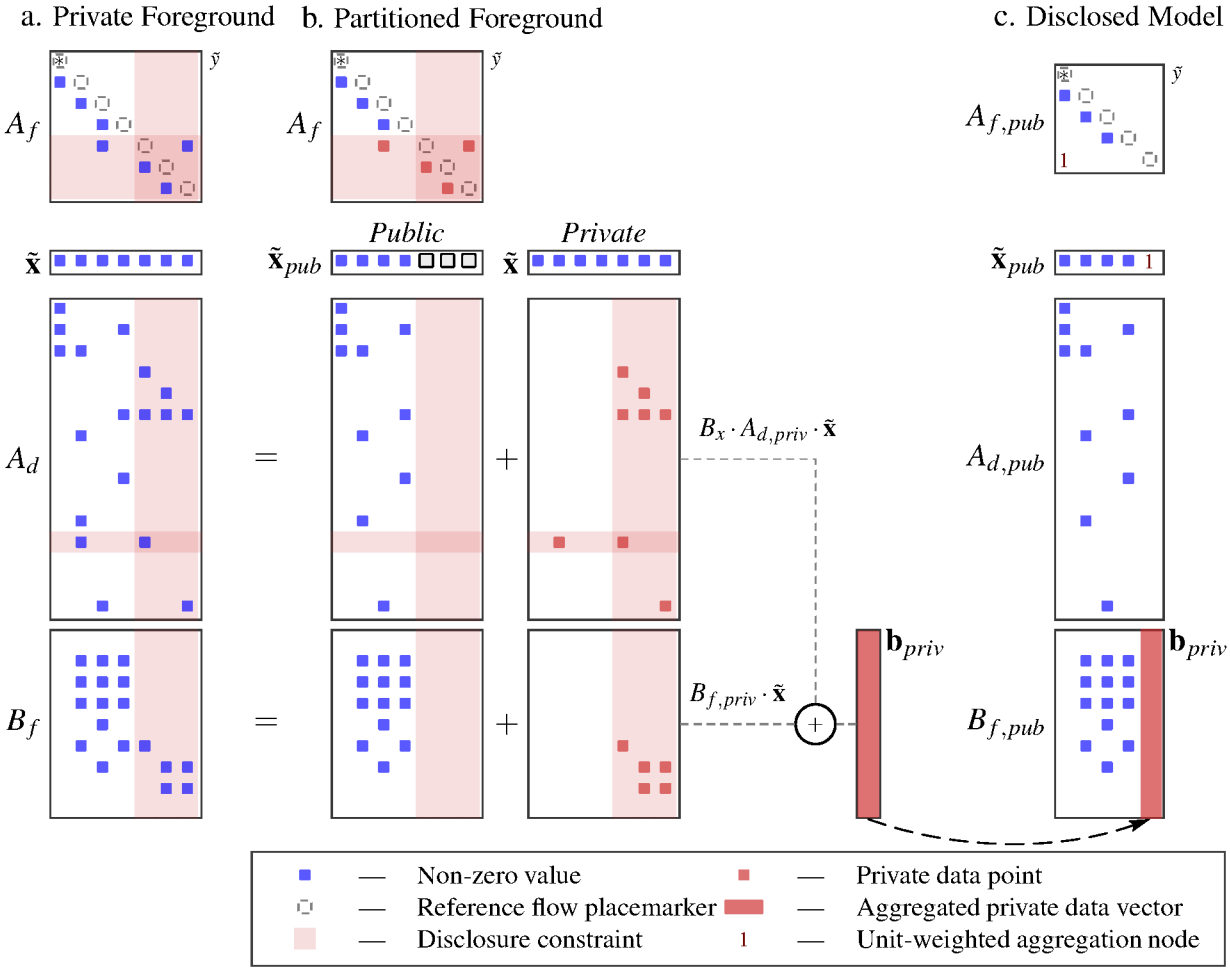
$$\mathbf{c}^T \cdot \left(\tilde{\mathbf{b}}_x + \tilde{\mathbf{b}}_f \right)$$



Graphical depiction of the components of a disclosure.

72x84mm (300 x 300 DPI)




a. Private Foreground
b. Partitioned Foreground
c. Disclosed Model
Public
Private
Non-zero value
Reference flow placemarker
Disclosure constraint
Private data point
Aggregated private data vector
Unit-weighted aggregation node

Disclosure of Product System Models in Life Cycle Assessment: Achieving Transparency and Privacy

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

2018-04-23

Supporting Information

1 List of Mathematical Symbols

The symbols used in the text are defined in Table S1.

2 Mathematical Relations

2.1 Final Demand Augmentation

This is a proof of Equation 2 in the manuscript.

Given a life cycle inventory database (LCIDB) containing input-output matrix A and emission matrix B, and a final demand vector $\mathbf{y}$, construct an augmented inventory database $\tilde{A} = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{y} & A \end{bmatrix}$, $\tilde{B} = [\mathbf{0}, B]$, and canonical functional unit $\tilde{\mathbf{y}} = [1, 0, 0, \ldots, 0]^T$. Show that

$$B \cdot (I - A)^{-1} \cdot \mathbf{y} = \tilde{B} \cdot \left(I - \tilde{A}\right)^{-1} \cdot \tilde{\mathbf{y}} \tag{1}$$

We begin by constructing $(I - \tilde{A})^{-1}$. Because the matrix is block triangular, we only need to determine the value of $\mathbf{q}$ in the matrix below:

$$I = (I - \tilde{A})^{-1} \cdot (I - \tilde{A}) \tag{2}$$

$$\begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & I \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{q} & (I - A)^{-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & \mathbf{0}^T \\ -\mathbf{y} & (I - A) \end{bmatrix} \tag{3}$$

Table S1: List of Mathematical Symbols

Symbol	Formula	Size	Description	ref.
Dimensions				
m		scalar	Number of emissions	Eq. 1
n		scalar	Number of process outputs in background database	Eq. 1
p		scalar	Number of nodes in foreground model	Fig. 2
p'		scalar	Number of cut-off nodes in foreground	Fig. 4c
N	$n+p$	scalar	size of augmented matrix	n.a.
Classical LCA				
A		$n \times n$	Direct requirements matrix (Leontief)	Eq. 1
$I-A$		$n \times n$	Technology matrix	"
B		$m \times n$	Emission matrix	"
$\mathbf{y}$		$n \times 1$	Final demand vector	"
$\mathbf{x}$	$(I-A)^{-1} \cdot \mathbf{y}$	$m \times 1$	Activity level vector	Fig. 1
$\mathbf{b}$	$B \cdot \mathbf{x}$	$m \times 1$	Life cycle inventory	"
$\mathbf{c}$		$m \times 1$	Characterization vector	Eq. 1
s	$\mathbf{c}^T \cdot \mathbf{b}$	scalar	LCIA Result	"
Product System Model				
$\tilde{\mathbf{y}}$	$[1,0,0,\ldots,0]^T$	$N \times 1$	Canonical functional unit vector	Eq. 2
$\tilde{A}$		$N \times N$	Augmented A matrix	Eq. 3
$\tilde{B}$		$m \times N$	Augmented B matrix	"
A_f		$p \times p$	Foreground matrix	"
A_d		$n \times p$	Background dependency matrix	"
B_f		$m \times p$	Foreground emission matrix	"
B_x	$B \cdot (I-A)^{-1}$	$m \times n$	Background LCI database matrix	Eq. 4
$\tilde{\mathbf{y}}_f$	$[1,0,0,\ldots,0]^T$	$p \times 1$	Canonical foreground functional unit	"
$\tilde{\mathbf{x}}$	$(I-A_f)^{-1} \cdot \tilde{\mathbf{y}}_f$	$p \times 1$	Foreground activity vector	Fig. 2
Foreground Aggregation				
$\tilde{\mathbf{b}}_f$	$B_f \cdot \tilde{\mathbf{x}}$	$m \times 1$	Aggregated foreground emissions	Eq. 5
$\tilde{\mathbf{a}}_d$	$A_d \cdot \tilde{\mathbf{x}}$	$n \times 1$	Aggregated dependencies	"
$\tilde{\mathbf{b}}_x$	$B_x \cdot \tilde{\mathbf{a}}_d$	$m \times 1$	Aggregated background	"
$\tilde{\mathbf{a}}_f$		$p' \times 1$	Vector of foreground cut-off flows	Fig. 4c
$\tilde{\mathbf{b}}$	$\tilde{\mathbf{b}}_x + \tilde{\mathbf{b}}_f$	$m \times 1$	Aggregated life cycle inventory	Fig. 4f
Private Aggregation				
$\tilde{\mathbf{a}}_{d,priv}$		$n \times 1$	Private dependencies	Eq. 7
$\tilde{\mathbf{a}}'_d$	$\tilde{\mathbf{a}}_d - \tilde{\mathbf{a}}_{d,priv}$	$n \times 1$	Disclosed dependencies	"
$A_{d,priv}$		$n \times p$	Private dependency matrix	Eq. 8
$A_{d,pub}$	$A_d - A_{d,priv}$	$n \times p$	Public dependency matrix	"
$B_{b,priv}$		$m \times p$	Private foreground emission matrix	"
$B_{f,pub}$	$B_f - B_{f,priv}$	$m \times p$	Public foreground emission matrix	"
$\mathbf{b}_{priv}$	$(B_{f,priv} + B_x \cdot A_{d,priv}) \cdot \tilde{\mathbf{x}}$	$m \times 1$	Private life cycle inventory	Eq. 9
φ	Eq. 12	scalar	Disclosure completeness metric	Eq. 12

It can be seen that $\mathbf{q} - (I - A)^{-1} \cdot \mathbf{y} = \mathbf{0}$ is a necessary condition to satisfy the equality, and thus $\mathbf{q} = (I - A)^{-1} \cdot \mathbf{y}$.

We know that $\tilde{\mathbf{y}}$ selects the first column of its argument, so:

$$(I - \tilde{A})^{-1} \cdot \tilde{\mathbf{y}} = \begin{bmatrix} 1 \\ \mathbf{q} \end{bmatrix} \tag{4}$$

And therefore

$$\begin{aligned} \tilde{B} \cdot (I - \tilde{A})^{-1} \cdot \tilde{\mathbf{y}} &= \begin{bmatrix} \mathbf{0} & B \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \mathbf{q} \end{bmatrix} && (5) \\ &= B \cdot \mathbf{q} && (6) \\ &= B \cdot (I - A)^{-1} \cdot \mathbf{y} && (7) \end{aligned}$$

thus completing the proof.

2.2 The Study Foreground Equation

This section contains the derivation of Equation 4 in the manuscript, in two steps.

2.2.1 "Flattening" the background database

First, information about the background database is removed from $\tilde{A}$ and replaced by an aggregated LCI database in a process I call "flattening."

Given a canonical LCA foreground study, show that the following formulations are equivalent:

$$\tilde{A} = \begin{bmatrix} A_f & 0 \\ A_d & A \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} B_f & B \end{bmatrix} \tag{8}$$

$$\tilde{A}_{flat} = \begin{bmatrix} A_f & 0 \\ A_d & 0 \end{bmatrix}; \quad \tilde{B}_{flat} = \begin{bmatrix} B_f & B_x \end{bmatrix} \tag{9}$$

where $B_x = B \cdot (I - A)^{-1}$.

It is sufficient to show that $\tilde{B} \cdot (I - \tilde{A})^{-1} = \tilde{B}_{flat} \cdot (I - \tilde{A}_{flat})^{-1}$.

Using the shorthand $Z = I - A$, we define the following identity:

$$I* = \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix}^{-1} \cdot \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix} \tag{10}$$

We introduce the identity in between the two terms of the LCIDB:

$$\tilde{B}\cdot(I-\tilde{A})^{-1} = \begin{bmatrix} B_f & B \end{bmatrix}\cdot I * \cdot \begin{bmatrix} I-A_f & 0 \\ -A_d & I-A \end{bmatrix}^{-1} \tag{11}$$

$$= \begin{bmatrix} B_f & B \end{bmatrix}\cdot \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix}^{-1} \cdot \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix} \cdot \begin{bmatrix} I-A_f & 0 \\ -A_d & Z \end{bmatrix}^{-1} \tag{12}$$

Using the associative property of matrix multiplication, it can be shown that $M\cdot N^{-1} = (N\cdot M^{-1})^{-1}$ for any invertible M and N of the same rank, so:

$$\tilde{B}\cdot(I-\tilde{A})^{-1} = \begin{bmatrix} B_f & B \end{bmatrix}\cdot \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} I-A_f & 0 \\ -A_d & Z \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix}^{-1} \right)^{-1} \tag{13}$$

$$= \begin{bmatrix} B_f & B \end{bmatrix}\cdot \begin{bmatrix} I & 0 \\ 0 & Z^{-1} \end{bmatrix} \cdot \left(\begin{bmatrix} I-A_f & 0 \\ -A_d & Z \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & Z^{-1} \end{bmatrix} \right)^{-1} \tag{14}$$

$$= \begin{bmatrix} B_f & B\cdot Z^{-1} \end{bmatrix}\cdot \left(\begin{bmatrix} I-A_f & 0 \\ -A_d & I \end{bmatrix} \right)^{-1} \tag{15}$$

But these terms are equivalent to the flattened LCIDB:

$$\tilde{B}\cdot(I-\tilde{A})^{-1} = \begin{bmatrix} B_f & B\cdot(I-A)^{-1} \end{bmatrix}\cdot \left(\begin{bmatrix} I-A_f & 0 \\ -A_d & I \end{bmatrix} \right)^{-1} \tag{16}$$

$$= \begin{bmatrix} B_f & B_x \end{bmatrix}\cdot \left(I - \begin{bmatrix} A_f & 0 \\ A_d & 0 \end{bmatrix} \right)^{-1} \tag{17}$$

$$= \tilde{B}_{flat}\cdot(I-\tilde{A}_{flat})^{-1} \tag{18}$$

thus completing the proof.

2.2.2 Separating Background from Foreground

Equation 4 in the manuscript is derived from Equation 31 above. It was also derived in Kuczenski (2015).

We begin by computing $(I-\tilde{A}_{flat})^{-1}$, again taking advantage of the matrix's block triangularity. Find Q such that:

$$\begin{bmatrix} (I-A_f) & 0 \\ -A_d & I \end{bmatrix} \cdot \begin{bmatrix} (I-A_f)^{-1} & 0 \\ Q & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{19}$$

It can be seen that $Q = A_d\cdot(I-A_f)^{-1}$ is a necessary condition to satisfy the equality, and thus:

$$(I-\tilde{A})^{-1} = \begin{bmatrix} (I-A_f)^{-1} & 0 \\ A_d\cdot(I-A_f)^{-1} & I \end{bmatrix} \tag{20}$$

.

Substituting this into the LCA system equation:

$$s = \mathbf{c}^T \cdot \tilde{B}_{flat} \cdot (I - \tilde{A}_{flat})^{-1} \tilde{\mathbf{y}} \tag{21}$$

$$= \mathbf{c}^T \cdot \begin{bmatrix} B_f & B_x \end{bmatrix} \cdot \begin{bmatrix} (I - A_f)^{-1} & 0 \\ A_d \cdot (I - A_f)^{-1} & I \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{y}}_f \\ \mathbf{0} \end{bmatrix} \tag{22}$$

where $\tilde{\mathbf{y}}_f$ is a canonical functional unit having the same dimension as A_f. Thus:

$$s = \mathbf{c}^T \cdot \begin{bmatrix} B_f & B_x \end{bmatrix} \cdot \begin{bmatrix} (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f \\ A_d \cdot (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f \end{bmatrix} \tag{23}$$

$$= \mathbf{c}^T \cdot \left(B_f \cdot (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f + B_x \cdot A_d \cdot (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f \right) \tag{24}$$

$$= \mathbf{c}^T \cdot \left(B_f + B_x \cdot A_d \right) \cdot (I - A_f)^{-1} \cdot \tilde{\mathbf{y}}_f \tag{25}$$

$$= \mathbf{c}^T \cdot \left(B_f + B_x \cdot A_d \right) \cdot \tilde{\mathbf{x}} \tag{26}$$

thus completing the derivation.

3 On the Equivalence of Process and Input-Output LCA

There is a level of confusion about the differing implications of process-based LCA inventory analysis as described in Heijungs and Suh (2002), and input-output analysis using a Leontief matrix. In fact, these two techniques were shown to be equivalent in Suh et al, 2010[1], whose abstract declares: "the article shows that the partitioning method in LCA is equivalent to the industry-technology model in input-output economics, and system expansion in LCA is equivalent to the by-product-technology model in input-output output economics." The article does indeed show convincingly that "the two data types can be hybridized without the loss of methodological consistency" (p.345).

Both the technology matrix A in LCA and the Leontief matrix $(I - A)$ in input output analysis are ways of encoding the relationships between the products that some processes "make" and other processes "use." In either the classical or input-output formulation, both matrices must be put into symmetric form to be invertible, and this means describing each column as having a single, reference output. This is done either through disaggregation (i.e. dividing a process by its various outputs), aggregation (i.e. combining products to form a market or industry), or some combination of these. While the task of transforming data is highly case-dependent, it is not constrained by the choice of mathematical formalism.

Suh et al, after a rich investigation of the literature, a thorough mathematical development, and a worked example, conclude that "even very large-scale LCI problems, such as those in commercial LCI databases, can be computed with a consistent mathematical framework…" that being the make-and-use framework (p. 349). In effect, the different mathematical approaches can be made equivalent. The equivalency can be made very plain by simply reinterpreting the input-output expressions *as* technology matrices. In other words, (in Suh et al) the expressions $(I - A_I)$ (eq. 6), $(I - A_C)$ (eq. 9), $(I - A_B)$ (eq. 12), and $(V' - U)$ (eq. 17) can simply be interpreted as *different*

[1] doi: 10.1111/j.1530-9290.2010.00235.x; see full text for citation.

ways to formulate the process-based technology matrix. There is no inconsistency between this interpretation and the classical formulation.

Once single-output processes are obtained, one substantive difference between the various approaches is column scaling: whether the entries describe real outputs, unit outputs, or unit activity levels. The database creator must ensure that a consistent scaling is applied to both the technology matrix and the emission matrix B. The convention in LCA is for a column to represent a unit activity level. To use the input-output coefficient approach, as in the present article, the columns must be scaled to a *unit of the reference product*. This reference output is then omitted from the A matrix, and is instead supplied by the I in the expression $(I - A)$. This formulation is completely consistent with any other matrix-based formulation of LCA, including classical process LCA as traditionally practiced. In particular, the approaches described here apply equally well to LCA models using both process-based databases such as Ecoinvent, input-output databases such as Exiobase, and background databases such as the GaBi databases.

Interested readers should note that the canonical reference for LCA, ISO 14044, has nothing to say about the mathematical techniques used to solve LCA problems, and the only preference for a technology matrix over an input-output matrix is borne of tradition or custom rather than theory. SimaPro, the leading worldwide software system for scientific LCA, converted to using an iterative Leontief inverse for Version 8, as disclosed in a whitepaper published in 2014[2], realizing significant speed and memory usage improvements. OpenLCA, the open-source alternative, also reported a dramatic reduction in memory usage and computation time around the release of version 1.4 in 2014, coincident with a revised algorithm "optimized for huge databases."[3] This is highly suggestive of a switch to an iterative approach. GaBi, the more industrially oriented software, does not use a matrix formulation at all, but has always computed activity levels using iteration at the plan level.

4 Meeting the Disclosure Objectives

4.1 Transparency

The objective for transparency was identified as requiring computability, completeness, and reproducibility. These requirements can be met by a disclosure that clearly describes the identities of each row or column of A_f, A_d, and B_f containing non-zero entries, and the locations and values of those entries. In principle, a reader with this information would have the capability to construct the augmented LCIDB and perform the computation in Eq. 4. In actuality, while it is easy to reproduce a set of sparse matrices, there is considerable potential for ambiguity in stating the identities of the rows and columns of those matrices.

The foreground nodes, which make up the columns of A_f, can be chosen freely by the study author according to the objectives of the study. Large studies may contain hundreds of foreground nodes, and the nodes can correspond to physical activities, logical operations, unit conversions,

[2] https://www.pre-sustainability.com/news/new-calculation-engine-simapro-8
[3] http://www.openlca.org/openlca-beta-1-4-released-optimised-for-huge-databases/

accumulation or distribution points, or any other aspect of model construction that can be reflected in a process-flow diagram. The only requirement on their disclosure is that the identity of each node's reference flow, including its unit of measure, is clearly stated.

For each row in A_d containing a non-zero entry, the author must unambiguously identify the exact dataset used, including the version of the database, as well as the exact process and reference flow selected; the dimension (reference quantity or unit) of each reference flow must be specified; and the sign of the numeric entry in A_d must be consistent with the implementation of the process in the background database. Similarly, for each row in B_f containing a non-zero entry, the author must unambiguously identify the substance being exchanged with the environment, the compartment or context into which it is being exchanged, and the reference quantity or unit associated with the flow. Sign consistency must also be assured. If LCIA indicator results are included, the author must also unambiguously identify the method computed (identity of $\mathbf{c}$).

4.2 Authority and Originality

It may be observed that many LCA studies make use of data sources that have been previously published but that have not been included in any LCI reference or background database. As long as these data sources can be integrated seamlessly into an LCA computation, it is not necessary to reproduce them in a disclosure. However, in most applications, previously published data must be re-implemented by the author in the LCA software context, and often this re-implementation requires a re-interpretation of the data source as a unit process inventory, where one exchange is recognized as a reference flow and other exchange values are reported in proportionality to the reference flow.

Because of these conditions, in the vast majority of cases the LCA study disclosure must include the author's reimplementations in order to achieve both the aims for authority and originality. In so doing, the disclosure enables a critical reviewer to evaluate whether the author's implementations are generally correct and complete.

In the future, the inclusion of external data automatically can be accomplished in the same way that reference LCI data could conceivably be included automatically: by the data providers making their information available using a stable semantic reference to a specialized Web-based application programming interface (API). This would have the benefits of enabling downstream users to access the information without having to re-implement it, thus reducing the size of the disclosure necessary to describe the PSM and simplifying the task of the modeler.

5 Preparing the Foreground Matrix

The development in the main text assumes the product system model is already described as a symmetric collection of single-output processes (activities) and flows (commodities) related by coefficients. In concrete terms, this means that:

- each node in the foreground (column shared by A_f, A_d, and B_f) corresponds to a particular flow in the model's foreground (row of A_f);
- a unit activity of each node in the foreground corresponds to a unit output of each flow;
- the coefficients in A_f, A_d, and B_f are ratios expressing the quantity of the flow (row), in its native unit, exchanged per unit of the reference flow (column).

Each foreground column has both an intrinsic reference flow and a unit of measure, and the use of this flow by other processes must be made with regard to this measure. The activity level of each foreground node necessarily equals the magnitude of the total reference flow emanating from the node. For background dependencies and emissions (rows of A_d and B_f, respectively), the modeler must take care to use the units established by the references, but in the foreground case the modeler must ensure to be self-consistent in preparing the A_f matrix.

5.1 Simple Foreground Networks

Figure S1 illustrates the structure of some basic foreground configurations. Most product system models would include many of these elements mixed together. All the activities in these examples are single-output. The first (a) is a sequential model, in which each node requires one foreground input and generates one output. This model is equivalent to a "gate to gate" model. Here the weights k_i indicate the amount of the preceding reference flow that is required by the subsequent node. Figure S1(b) shows an additive model, in which the outputs of several foreground nodes are added together, equivalent to a "mixer" or a horizontal average. In this arrangement the weights should add up to a unit output of the reference node. Figure S1(c) shows a foreground model with a cyclic dependency, where some of the reference output is consumed by another foreground node.

A typical PSM may contain multiple modules or fragments that are interconnected. An example of foreground composed of several fragments is illustrated in Figure S1(d). Here, the nodes labeled 1–5 represent one fragment, which generates the foreground's canonical reference flow $\tilde{y}$. This fragment requires two interior flows from separate fragments (y_0 and y_1), and has two unconnected flows (4 and 5). The reference y_0 is supplied by a second fragment, made up of nodes 6-8. The reference y_1 is supplied by another fragment made of only one node (9). The reference flow y_1 is consumed in two different places by the other fragments.

As noted, a reviewer with access to items *d-i* and *d-iv* in the disclosure would be able to construct A_f and automatically create a process-flow diagram, annotated with information about each node.

5.2 Multi-Output Foreground Processes

The treatment of multi-functional activities in LCA is a topic of constant discussion among practitioners and theorists. While the question of the correct, appropriate, or ideal method of apportioning impacts among co-products cannot be resolved in general, the utility of a disclosure framework must be to *express* transparently how co-production was resolved within a given foreground model. Majeau-Bettez et al. (2014) recently provided a general framework (referred to hereafter

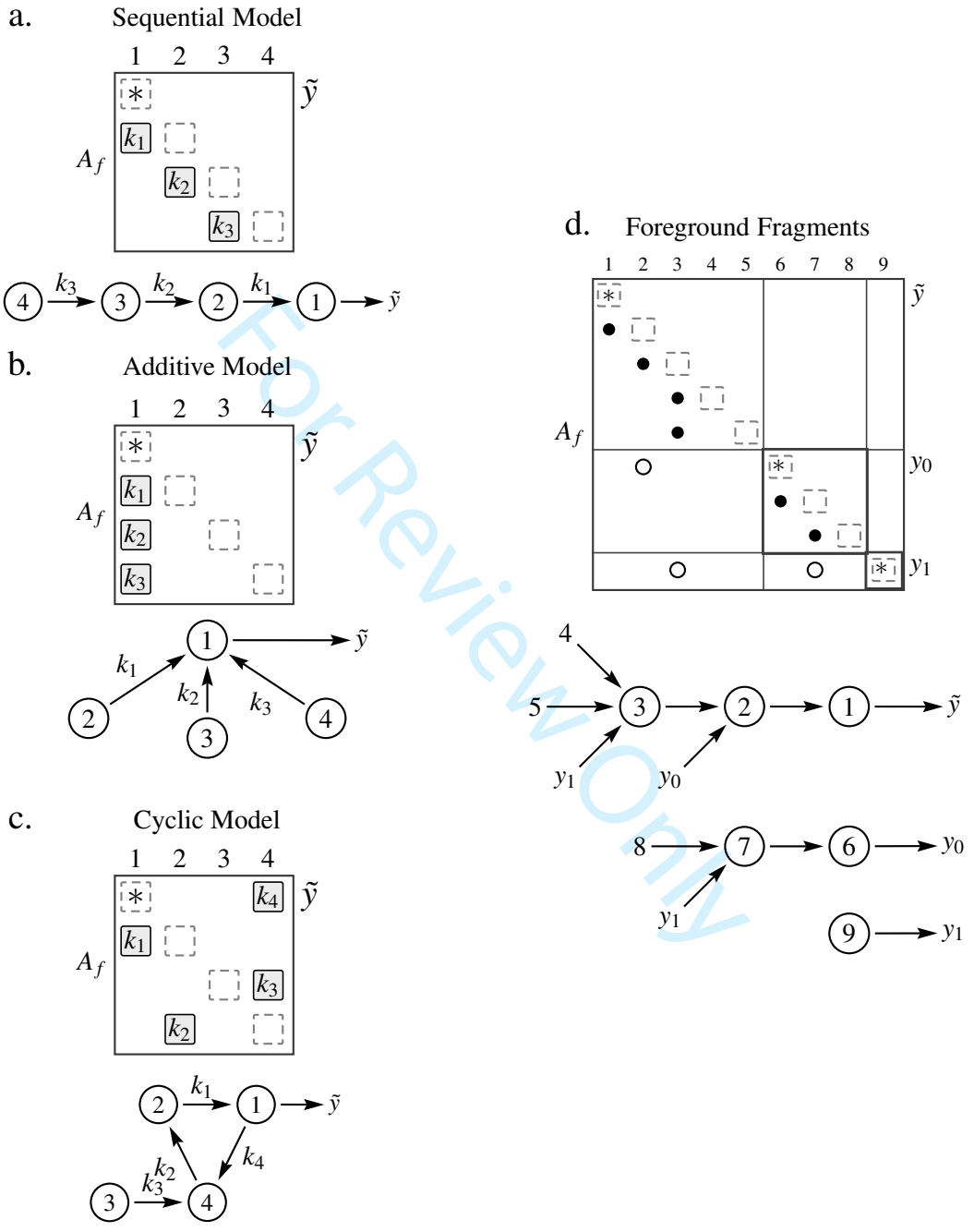


Figure S1: Equivalent matrix representations and graphs for different foregrounds.

Table S2: Example: Chlor-alkali electrolysis, membrane cell [RER] (Ecoinvent 3.2; excerpt)

Flow	Unit	Direction	Value
Reference:			
Chlorine, gaseous	kg	Output	1
Hydrogen, liquid	kg	Output	0.028
Sodium hydroxide	kg dry	Output	1.13
Requirements:			
Electricity, medium voltage	kWh	Input	2.97
Sodium chloride, powder	kg	Input	1.75
Chemical factory, organics	unit	Input	4e-10
Sludge, NaCl electrolysis	kg	Output	0.0153
Emissions:			
Chloride [16887-00-6], surface water	kg	Output	0.015
Carbon dioxide [124-38-9], urban air	kg	Output	0.0031

as MWS14) for describing the treatment of co-products, and I adopt that framework here to show how the technique used to handle co-production can be disclosed transparently for foreground processes.

5.2.1 Example System

In all strategies, an abbreviated example system drawn from Ecoinvent, reporting chlor-alkali electrolysis using a membrane cell, is used. The un-allocated system is shown in Table S2.

The objective is to disclose explicitly how the multi-output system was resolved to single-output activities for the purposes of review, reproduction, or modification. Therefore, the raw (un-allocated) process inventory is included in all cases. Examples are provided for partitioning allocation, classical system expansion, and consequential system expansion.

5.2.2 Partitioning Allocation

According to MWS14, Partitioning allocation (PA) is a "production-balanced model that artificially splits the requirements of an activity across its different outputs." The exact manner of the partitioning need not be specified, as long as the parts sum up to the whole. In the disclosure framework, any manner of production-balanced partitioning can be represented by including the raw, un-allocated inventory as one foreground column, choosing any output as the reference. The other $n-1$ outputs are then included as allocated inventories in additional columns, for a total of n columns. Each column must be normalized to a unit reference.

Note that, although it is customary for allocated process inventories to be derived as a uniform partition of the un-allocated process based on some parameter such as mass, price or energy content, this is not necessary. For a process with n outputs the modeler has $n-1$ degrees of freedom

Table S3: Chlor-alkali electrolysis as a foreground model, partitioning allocation

A_f	0	1	2
Chlorine, gaseous (kg)	*		
Hydrogen, liquid (kg)	-0.028	*	
Sodium hydroxide (kg dry)	-1.13		*
A_d	0	1	2
Electricity, medium voltage (kWh)	2.97	1.38	1.38
Sodium chloride, powder (kg)	1.75	0.811	0.811
Chemical factory, organics (unit)	4e-10	1.9e-10	1.9e-10
Sludge, NaCl electrolysis (kg)	0.0153	0.00709	0.00709
B_f	0	1	2
Chloride, to surface water (kg)	0.015	0.00695	0.00695
Carbon dioxide, to urban air (kg)	0.0031	0.00144	0.00144

in specifying the allocated inventories, with the requirement that the nth inventory equals the remainder. In the framework presented, the modeler must provide the $n-1$ allocated inventories explicitly. The reference flow for the nth process will simply be computed as the remainder when the allocated inventories are subtracted from the un-allocated inventory.

Table S3 illustrates this for the chlor-alkali example, for conventional allocation by mass. The unaltered inventory is shown in column 0, corresponding to the reference product of gaseous chlorine. Co-products are shown as additional foreground flows, with negative signs indicating that the flows are outputs. The reference for each column is indicated by an asterisk.

The inventories themselves are shown as entries in the A_d and B_f matrices. Because all co-products are already reported in mass, the allocated inventories for the two co-products are both the same and represent a scaling-down of the un-allocated inventory by the total mass of outputs (about 2.16 kg).

Using the foreground in Table S3, LCA computation using the canonical functional unit (see main text, Eq. 3) will automatically compute the self-consistent single-output result for 1 kg of chlorine gas because the allocated inventories for hydrogen and sodium hydroxide will be subtracted from the total. The foreground activity levels for the system are:

$$\tilde{\mathbf{x}} = [1, -0.028, -1.13]^T \tag{27}$$

with the resulting aggregated dependency and emission vectors being, again, the same as the normalized and allocated inventories:

$$\tilde{\mathbf{a}}_d = A_d \cdot \tilde{\mathbf{x}} = \quad [1.38, 0.811, 1.9 \cdot 10^{-10}, 0.00709]^T \tag{28}$$

$$\tilde{\mathbf{b}}_f = B_f \cdot \tilde{\mathbf{x}} = \quad [0.00695, 0.00144]^T \tag{29}$$

At the same time, LCA computation using a functional unit of one of the co-products will simply

make use of the allocated inventory. In this example, all processes are identical on a mass basis, but any other partitioning allocation can also be automatically rendered self-consistent through this approach. A reviewer would be able to inspect the foreground model to determine whether the allocation was performed as described in the report.

5.2.3 "Classical" System Expansion

Although the term 'system expansion' is often used to refer to product substitution and/or alternate activity allocation, these modeling techniques are distinct and should be referred to with different names, as noted in GWS14. In the original definition of system expansion, the functional unit is simply enlarged to include all co-products. Representation of such a system is trivial and is shown in Table S4.

Table S4: Chlor-alkali electrolysis as a foreground model, classical system expansion

A_f	0	1	2
Chlorine, gaseous (kg)	*		
Hydrogen, liquid (kg)	-0.028	*	
Sodium hydroxide (kg dry)	-1.13		*
A_d	0	1	2
Electricity, medium voltage (kWh)	2.97		
Sodium chloride, powder (kg)	1.75		
Chemical factory, organics (unit)	4e-10		
Sludge, NaCl electrolysis (kg)	0.0153		
B_f	0	1	2
Chloride, to surface water (kg)	0.015		
Carbon dioxide, to urban air (kg)	0.0031		

In the classical system expansion case, applying the expanded functional unit of 1 kg chlorine, plus 0.028 kg hydrogen and 1.13 kg of sodium hydroxide, will result in a unit activity of the reference process, without any cut-off flows. Equivalently, the canonical functional unit, production of 1 kg of chlorine gas, will result in the same process activity level, and will also generate the co-products as cutoff flows.

5.2.4 Consequential System Expansion

The foreground disclosure framework is also well suited to describe consequential system expansion through product substitution or "alternate activity allocation," which are the terms used in MWS14. An example illustrating these methods is shown in Table S5

Here an alternate activity, namely electrolysis of water, is included for liquid hydrogen production and is specified at index 3 in the foreground (shown in blue). The inventory converts 9 kg of

Table S5: Chlor-alkali electrolysis as a foreground model, consequential system expansion. An inventory for the electrolysis of water (shown in blue) is provided as an alternate activity to produce hydrogen, while the sodium hydroxide product is assumed to substitute for for magnesium hydroxide (shown in red).

A_f	0	1	2	*3*
Chlorine, gaseous (kg)	*			
Hydrogen, liquid (kg)	-0.028	*		
Sodium hydroxide (kg dry)	-1.13		*	
Hydrogen, electrolysis (kg)		*1*		*
A_d	0	1	2	*3*
Electricity, medium voltage (kWh)	2.97			*50*
Sodium chloride, powder (kg)	1.75			
Chemical factory, organics (unit)	4e-10			
Sludge, NaCl electrolysis (kg)	0.0153			
Water, distilled (kg)				*9*
Magnesium hydroxide (kg dry)			*0.8*	
B_f	0	1	2	*3*
Chloride, to surface water (kg)	0.015			
Carbon dioxide, to urban air (kg)	0.0031			

water to 1 kg of hydrogen using 50 kWh of electricity. The hydrogen produced from the multi-output reference process is (somewhat dubiously) taken to displace hydrogen produced through the alternate activity by introducing a unity coefficient in row 3, column 1 of A_f.

Meanwhile, sodium hydroxide from the reference process is modeled to substitute the production of magnesium hydroxide through the introduction of a new row and a new nonzero coefficient in column 2 of A_d (shown in red). Here the match is inexact, and because magnesium hydroxide is more effective in some applications, 1 kg of sodium hydroxide is taken to displace only 0.8 kg of magnesium hydroxide.

6 Examples

LCI databases often contain product systems that can be modeled as foreground studies because they describe products that are not required by the background. In this section, two product systems selected from LCI databases are used to illustrate the concept of structured publication. The systems selected because are complex enough to illustrate the premise but simple enough to review easily.

Each system is illustrated as a table that shows the foreground model, cutoff flows, background dependencies and foreground emissions included in the system. Aggregation results $\tilde{\mathbf{x}}$, $\tilde{\mathbf{a}}_d$, and $\tilde{\mathbf{b}}_f$ are also reported. The table omits numeric data from most of the table for clarity. Instead, a black square indicates the presence of a nonzero value.

6.1 US LCI: Secondary Aluminum

The US LCI database contains a small background system of 39 processes, as well as 395 foreground product systems that range in size from one to 83 foreground flows.

The example product system reports the production of secondary aluminum from automotive scrap (Figure S2). The aluminum production process has direct requirements for two other foreground systems, including transportation services and quicklime production. The quicklime production itself depends on limestone extraction, which is also part of the foreground. The aluminum production requires two background systems, electricity and natural gas combustion, and reports 10 cutoff flows. The main input to the process, "Aluminum scrap, automotive," appears as a cutoff flow. Overall, the four foreground processes require the background for only combustion (five fuels), transport (three modes) and grid electricity. The same natural gas combustion and electricity models were used in all three non-transport foreground nodes.

The system also reports several direct emissions, mainly from the aluminum process. Only one resource consumption (input flow) is reported – the extraction of limestone. The direct emission modeling is limited in scope and includes a number of "unspecified" flows that may not be well characterized in many impact methods.

6.2 Ecoinvent: Organic Potatoes

Ecoinvent version 3 is provided in three different system models that reflect different linking strategies. The example is drawn from the "Allocation at the point of substitution" or APOS model of Ecoinvent 3.2, which includes 11,420 processes that produce 12,966 product flows. Of these, 10,282 are background flows and the rest are foreground flows.

The example system reports production of organic potatoes supplied to the global market (Figure S3). The foreground includes nine nodes, of which six (nodes 3 through 8) contain cyclic dependencies associated with the production of potato seeds. The Ecoinvent database's use of "markets" as mixer processes is evident in the table: node 0 is a mixer process that combines Swiss ("CH") production (node 1) weighted at 2.4% with rest-of-World ("RoW") production (node 2) weighted at 97.6%. That same market split (2.4 / 97.6) can also be seen in nodes 4 and 8.

Nodes 5 and 6, which make seeds ready "for setting," are mixed by node 8. Although nodes 5 and 6 are geographically distinguished (CH vs RoW), both can be seen to consume potato seed from the global market.

Looking at the dependency and emission lists, the different "signatures" of different kinds of processes can be seen: nodes 1, 2, 3, and 7 are clearly agricultural processes that require irrigation, tillage, manure and so on. Nodes 0 and 4 are visible as market processes, their only requirements being transport processes. Nodes 5 and 6 each consume electricity and require use of a "multi-storey building." Taken together, the CH-locality processes appear to use CH irrigation and electricity supply but are otherwise similar to their RoW counterparts. The product model includes no cutoff flows.

Aluminum, secondary, ingot, from automotive scrap, at plant [RNA]

(node) Foreground flows – A_f	0	1	2	3	
(0) Aluminum, secondary, ingot, from automotive scrap, at plant [RNA] (kg)	◌				
(1) Aluminum recovery, transport, to plant [RNA] (kg)	1.03	◌			
(2) Quicklime, at plant [RNA] (kg)	2.35e-05		◌		
(3) Limestone, at mine [RNA] (kg)			1.87	◌	
Foreground Node Weights $\tilde{x}$	1	1.03	2.35e-05	4.39e-05	
Input: CUTOFF Disposal, solid waste, unspecified, to sanitary landfill [CUTOFF Flows] (kg)	■		■		0.0842
Input: CUTOFF Filter media, at plant [CUTOFF Flows] (kg)	■				4.79e-05
Input: CUTOFF Lube oil, at plant [CUTOFF Flows] (kg)	■				8.75e-07
Input: CUTOFF Treatment gases, unspecified, at plant [CUTOFF Flows] (kg)	■				0.003
Input: CUTOFF Alloying additives, at plant [CUTOFF Flows] (kg)	■				0.021
Input: CUTOFF Chemicals, unspecified, used for wastewater treatment [CUTOFF Flows] (kg)	■				0.008
Input: CUTOFF Grain refiners, at plant [CUTOFF Flows] (kg)	■				0.0003
Input: CUTOFF Treatment salts, unspecified, at plant [CUTOFF Flows] (kg)	■				0.009
Input: CUTOFF Packaging, unspecified, at plant [CUTOFF Flows] (kg)	■				1.67e-05
Input: CUTOFF Aluminum scrap, automotive [CUTOFF Flows] (kg)	■				1.03

Background Dependencies – A_d	0	1	2	3	$\tilde{a_d}$
Transport, barge, average fuel mix [RNA] (t*km)			■		5.66e-07
Liquefied petroleum gas, combusted in industrial boiler [RNA] (l)			■		7.57e-10
Transport, combination truck, diesel powered [RNA] (t*km)		■	■		0.374
Diesel, combusted in industrial boiler [RNA] (l)			■	■	4.79e-08
Natural gas, combusted in industrial boiler [RNA] (m3)	■		■	■	0.223
Transport, train, diesel powered [RNA] (t*km)		■	■		0.0415
Electricity, at grid, US, 2000 [RNA] (kWh)	■		■	■	0.668
Bituminous coal, combusted in industrial boiler [RNA] (kg)			■	■	4.04e-06
Gasoline, combusted in equipment [RNA] (l)				■	2.25e-09

Foreground Emissions – B_f	0	1	2	3	$\tilde{b_f}$
Output: Lead [air, unspecified] (kg)	■				2.16e-07
Output: Suspended solids, unspecified [water, unspecified] (kg)	■				2.56e-05
Output: Particulates, unspecified [air, unspecified] (kg)	■		■	■	1.78e-07
Output: NMVOC, non-methane volatile organic compounds [air, unspecified] (kg)	■				4.7e-05
Output: BOD5, Biological Oxygen Demand [water, unspecified] (kg)	■				1.35e-09
Output: COD, Chemical Oxygen Demand [water, unspecified] (kg)	■				8.3e-07
Output: Sulfur dioxide [air, unspecified] (kg)			■		3.52e-09
Output: Dissolved solids [water, unspecified] (kg)	■				2.38e-08
Output: carbon dioxide [air, unspecified] (kg)			■		1.8e-05
Input: Limestone [resource, ground-] (kg)				■	4.39e-05
Output: Heavy metals, unspecified [water, unspecified] (kg)	■				2.25e-05
Output: Organic substances, unspecified [water, unspecified] (kg)	■				1.29e-06
Output: Acids, unspecified [air, unspecified] (kg)	■				4.31e-05

Figure S2: A structured product system model for secondary aluminum, drawn from US LCI. Exchange values are replaced with black squares for clarity.

market for potato, organic [GLO]

(node) Foreground flows – A_f	0	1	2	3	4	5	6	7	8	
(0) potato, organic [GLO] (kg)	◌									
(1) potato, organic [CH] (kg)	0.024	◌								
(2) potato, organic [RoW] (kg)	0.976		◌							
(3) potato seed, organic, at farm [CH] (kg)				◌	0.024					
(4) potato seed, organic, at farm [GLO] (kg)					◌	1	1			
(5) potato seed, organic, for setting [RoW] (kg)						◌			0.976	
(6) potato seed, organic, for setting [CH] (kg)							◌		0.024	
(7) potato seed, organic, at farm [RoW] (kg)					0.976			◌		
(8) potato seed, organic, for setting [GLO] (kg)		0.11	0.11	0.16				0.16	◌	
Foreground Node Weights $\tilde{x}$	1	0.024	0.976	0.00315	0.131	0.128	0.00315	0.128	0.131	

Background Dependencies – A_d	0	1	2	3	4	5	6	7	8	$\tilde{a_d}$
building, multi-storey [GLO] (m3)						■	■			5.24e-06
electricity, low voltage [GLO] (kWh)						■				0.00933
potato haulm cutting [GLO] (ha)		■	■	■				■		5.04e-05
tillage, hoeing and earthing-up, potatoes [GLO] (ha)		■	■	■				■		0.000101
transport, tractor and trailer, agricultural [GLO] (metric ton*km)		■	■	■				■		0.00113
transport, freight, light commercial vehicle [GLO] (metric ton*km)	■				■					0.0258
tillage, ploughing [GLO] (ha)		■	■	■				■		5.04e-05
green manure, organic, until March [GLO] (ha)		■	■	■				■		5.04e-05
transport, freight, lorry, unspecified [GLO] (metric ton*km)	■				■					0.404
transport, freight train [GLO] (metric ton*km)	■				■					0.138
transport, freight, inland waterways, barge [GLO] (metric ton*km)	■				■					0.0857
copper oxide [GLO] (kg)		■	■	■				■		0.000107
tillage, harrowing, by spring tine harrow [GLO] (ha)		■	■	■				■		5.04e-05
potato planting [GLO] (ha)		■	■	■				■		5.04e-05
solid manure loading and spreading, by hydraulic loader and spreader [GLO] (kg)		■	■	■				■		0.717
potato grading [GLO] (kg)		■	■	■				■		1.13
tillage, harrowing, by rotary harrow [GLO] (ha)		■	■	■				■		5.04e-05
harvesting, by complete harvester, potatoes [GLO] (ha)		■	■	■				■		5.04e-05
transport, freight, sea, transoceanic ship [GLO] (metric ton*km)	■				■					0.525
liquid manure spreading, by vacuum tanker [GLO] (m3)		■	■	■				■		0.000565
electricity, low voltage [CH] (kWh)							■			0.00023
irrigation [GLO] (m3)			■					■		0.0173
tillage, currying, by weeder [GLO] (ha)		■	■	■				■		0.000101
application of plant protection product, by field sprayer [GLO] (ha)		■	■	■				■		0.000222
irrigation [CH] (m3)		■		■						0.000426

Foreground Emissions – B_f	0	1	2	3	4	5	6	7	8	$\tilde{b_f}$
Input: Occupation, construction site [natural resource, land] (m2*year)						■	■			2.1e-06
Input: Transformation, from unspecified [natural resource, land] (m2)						■	■			1.05e-06
Input: Transformation, to industrial area [natural resource, land] (m2)						■	■			1.05e-06
Input: Energy, gross calorific value, in biomass [natural resource, biotic] (MJ)		■	■	■				■		3.87
Output: Phosphate [water, ground-] (kg)		■	■	■				■		6.25e-06
Output: Cadmium, ion [water, ground-] (kg)		■	■	■				■		1.14e-09
Output: Zinc, ion [water, ground-] (kg)		■	■	■				■		1.35e-06
Output: Nitrate [water, ground-] (kg)		■	■	■				■		0.00696
… (31 rows omitted)										

Figure S3: A structured product system model for organic potato production, drawn from Ecoinvent v3.2 (APOS). Exchange values are replaced with black squares for clarity.