unfold: removing the barriers to sharing and reproducing prospective life-cycle assessment databases

Romain Sacchi [1]

1 Paul Scherrer Institute, Villigen, Switzerland

DOI: 10.21105/joss.05198

Software

- Review
- Repository
- Archive

Editor: Frauke Wiese

Reviewers:

- @mfastudillo
- @MaximeAgez

Submitted: 05 January 2023
Published: 29 March 2023

Summary

unfold is a Python package that allows reproducing life-cycle databases which partially build on a data source that cannot be shared. It produces data packages that contain the differences between the databases to share and the licensed data source. The data package also includes a metadata file describing the databases, the author, and other helpful information. unfold allows one to pack and unpack any databases from a single data package to ease the sharing and reproducibility of prospective or scenario-based life-cycle assessment databases.

Statement of need

Life-cycle assessment (LCA) consists of quantifying the environmental footprint of a product or a process by considering impacts along each of its relevant life-cycle phases (ISO, 2006). It relies on databases describing thousands of industrial processes, which interdependency and interactions with the greater environment form large input-output matrices. However, these matrices – often modified to fit the need of a study – are difficult to share when they contain data covered by a restrictive End-user license agreement, such as that of the ecoinvent database (Wernet et al., 2016).

The difficulty in sharing these databases prevents the reproducibility of results and impedes the scientific validity of the work produced by the LCA community.

Using the LCA framework brightway2 (Mutel, 2017), unfold allows sharing modified LCA databases without:

- exposing the data under license,
- going through the steps of data modification,

but sharing instead data packages used to reproduce the databases locally (provided the End-users also have access to the licensed data source).

unfold is initially conceived to share heavily modified LCA databases, such as those generated with the premise package (Sacchi et al., 2022) or those regularly produced within the field of prospective LCA (where the need to modify the source database extensively is often required) – see the work of Cox (Cox et al., 2018), Mendoza (Mendoza Beltran et al., 2018), Joyce (Joyce & Björklund, 2022) and colleagues.

Description

LCA relies on input-output matrices representing product and service exchanges that are typically sparse, with a density inferior to 1%. Sometimes, for the need of sensitivity or scenario analysis, up to hundreds of thousands of exchanges are added, modified, or removed from the technosphere matrix, potentially reflecting significant changes in certain industrial sectors. (Mendoza Beltran et al., 2018) gives an example where exchanges related to power generation are modified throughout the database to reflect changes in generation efficiencies, regional electricity mixes, etc.

`unfold` calculates the scaling factors for each modified exchange with reference to its value before modification (i.e., the reference database) and stores them into a data package.

When another user reads the data package, the scaling factors are multiplied by the exchange value found in the local reference database to back-calculate the value of the modified exchange.

Software architecture

The example illustrated in Figure 1 represents a use case where User 1 (in blue) would like to share databases A and B, which are initially derived from database C – which cannot be shared. `unfold` creates a data package (zip file) where the difference across exchanges in databases A and B relative to the reference database C are stored as scaling factors. Additional metadata (e.g., author, description) is included in the data package, per Frictionless's specifications (Walsh & Pollock, 2022). User 2 (in green) reads the data package using `unfold` on its local computer, which back-calculates the value of each exchange by multiplying the original values found in the reference database with their corresponding scaling factors. This assumes User 2 also has local access to the reference database C.

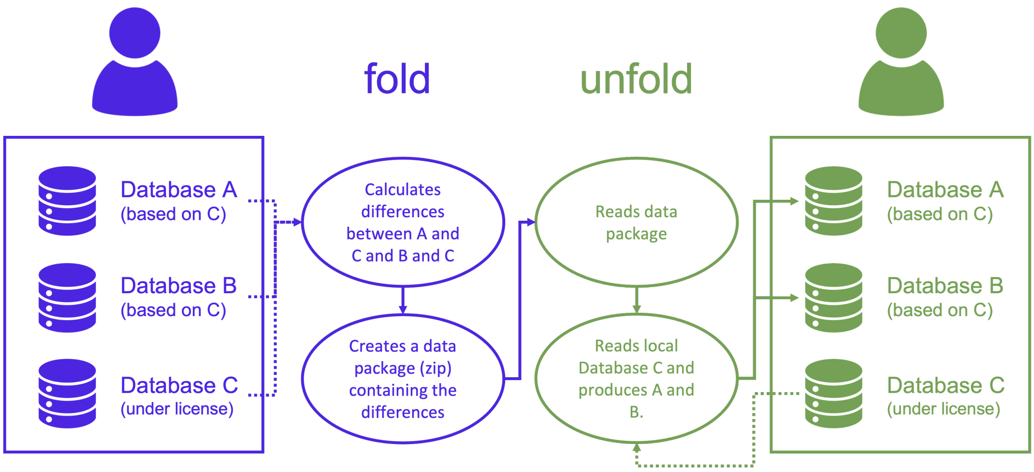

Figure 1: Workflow for sharing databases using `unfold` data packages.

Software functionalities

`unfold` offers two functions: pack and unpack LCA databases. The packing function calculates the differences between one or several modified databases and a reference database and stores them in a data package ready to be shared. The unpacking function creates and registers one or several LCA databases by reading the data package and the local reference database.

`unfold` can also produce superstructure databases (Steubing & Koning, 2021), which can integrate several scenarios into a single `brightway2` database, to be further used with the LCA software `activity-browser` (Steubing et al., 2020).

Impact

`unfold` allows sharing LCA databases that partially build on proprietary data without exposing the data under license. This eases the process of reproducibility, which is an essential prerequisite to scientific validity.

Sharing scenario-based or prospective LCA databases is already possible through other means, such as exchanging `Futura`'s recipes (Joyce & Björklund, 2022) or `premise`'s input parameters (Sacchi et al., 2022). However, the diversity of shared databases is eventually limited by the range of tools that generate them – most of the existing tools focus on modifying energy-intensive sectors. On the other hand, `unfold` allows sharing databases regardless of the tool or method used to generate them, as long as they present differences with the reference database. For example, a database with manually adjusted leakage rates of ozone-depleting gases can be shared just as easily.

Doing so allows sharing of the work of the different authors of these databases to be reused within the community. To that effect, a repository that lists several data packages already exists (Sacchi, 2022), allowing users to reproduce and use prospective LCA databases without executing the steps or operating the tools that led to their generation.

Conclusions

`unfold` offers a way to exchange and reproduce LCA databases that build partially on licensed data by sharing scaling factors in a data package.

Acknowledgements

Financial support was provided by the Kopernikus Project Ariadne (FKZ 03SFK5A), funded by the German Federal Ministry of Education and Research

References

Cox, B., Mutel, C. L., Bauer, C., Mendoza Beltran, A., & Vuuren, D. P. van. (2018). Uncertain Environmental Footprint of Current and Future Battery Electric Vehicles. *Environmental Science & Technology*, *52*(8), 4989–4995. https://doi.org/10.1021/acs.est.8b00261

ISO. (2006). *Environmental management — Life cycle assessment — Principles and framework* (pp. 1–28). ISO. https://doi.org/10.1136/bmj.332.7550.1107

Joyce, P. J., & Björklund, A. (2022). Futura: A new tool for transparent and shareable scenario analysis in prospective life cycle assessment. *Journal of Industrial Ecology*, *26*(1), 134–144. https://doi.org/10.1111/JIEC.13115

Mendoza Beltran, A., Cox, B., Mutel, C., Vuuren, D. van, Vivanco, D. F., Deetman, S., Edelenbosch, O., Guinée, J., & Tukker, A. (2018). When the Background Matters: Using Scenarios from Integrated Assessment Models in Prospective Life Cycle Assessment. *Journal of Industrial Ecology*. https://doi.org/10.1111/jiec.12825

Mutel, C. (2017). Brightway: An open source framework for Life Cycle Assessment. *The Journal of Open Source Software*, *2*(12), 236. https://doi.org/10.21105/joss.00236

Sacchi, R. (2022). *REpository for Prospective LIfe Cycle Assessment daTabasEs* [PhD thesis]. https://github.com/polca/replicate/blob/main/README.md

Sacchi, R., Terlouw, T., Siala, K., Dirnaichner, A., Bauer, C., Cox, B., Mutel, C., Daioglou, V., & Luderer, G. (2022). PRospective EnvironMental Impact asSEment (premise): A streamlined approach to producing databases for prospective life cycle assessment using integrated assessment models. *Renewable and Sustainable Energy Reviews*, *160*, 112311. https://doi.org/10.1016/j.rser.2022.112311

Steubing, B., & Koning, D. de. (2021). Making the use of scenarios in LCA easier: the superstructure approach. *The International Journal of Life Cycle Assessment*, *26*(11), 2248–2262. https://doi.org/10.1007/s11367-021-01974-2

Steubing, B., Koning, D. de, Haas, A., & Mutel, C. L. (2020). The Activity Browser — An open source LCA software building on top of the brightway framework. *Software Impacts*, *3*, 100012. https://doi.org/10.1016/j.simpa.2019.100012

Walsh, P., & Pollock, R. (2022). *Data Package | Frictionless Standards*. https://specs.frictionlessdata.io/data-package/#metadata

Wernet, G., Bauer, C., Steubing, B., Reinhard, J., Moreno-Ruiz, E., & Weidema, B. (2016). The ecoinvent database version 3 (part I): overview and methodology. *The International Journal of Life Cycle Assessment*, *21*(9), 1218–1230. https://doi.org/10.1007/s11367-016-1087-8