Industrial Ecology Freiburg
WORKING PAPER
1
2024

Guidelines and Good Practice Examples for Complete Traceability of Workflows and Reproducibility of Results in Industrial Ecology Research

Stefan Pauliuk, Christoph Helbig, Richard C Lupton, Peter Paul Pichler, Simon Schulte, Konstantin Stadler, Peng Wang, and Dominik Wiedenhofer

Endorsed by the Board of the Topical Section for Research on Socio-Economic Metabolism (SEM) of the International Society for Industrial Ecology (ISIE).

Freiburg, Germany, July 2024

IEF – Industrial Ecology Freiburg
Research Group for Industrial Ecology
Faculty of Environment and Natural Resources
University of Freiburg, Germany.

Please cite as follows:

Guidelines and Good Practice Examples for Complete Traceability of Workflows and Reproducibility of Results in Industrial Ecology Research. Stefan Pauliuk, Christoph Helbig, Richard C Lupton, Peter Paul Pichler, Simon Schulte, Konstantin Stadler, Peng Wang, and Dominik Wiedenhofer. Industrial Ecology Freiburg (IEF) Working Paper 1(2024), University of Freiburg, Germany, DOI 10.6094/UNIFR/255618

Contact info:

The author can be contacted via in4mation@indecol.uni-freiburg.de

Permalinks to this document:

https://doi.org/10.6094/UNIFR/255618

Industrial Ecology Freiburg (IEF) Working Papers is a series of scientific reports by the research group for industrial ecology at the Faculty of Environment and Natural Resources, University of Freiburg, Germany.

For more info, visit https://www.indecol.uni-freiburg.de/en

Guidelines and Good Practice Examples for Data Provenance and Traceability in Industrial Ecology Research

July 2024

Stefan Pauliuk,[1] Christoph Helbig,[2] Richard (Rick) C Lupton,[3] Peter Paul Pichler,[4] Simon Schulte,[1] Konstantin Stadler,[5] Peng Wang[6], and Dominik Wiedenhofer[7]

[1]) Industrial Ecology Group, Faculty of Environment and Natural Resources, University of Freiburg, Germany

[2]) Faculty of Engineering Science, University of Bayreuth, Germany

[3]) Institute for Sustainability, University of Bath, UK

[4]) Potsdam Institute for Climate Impact Research, Germany

[5]) Industrial Ecology Program, Norwegian University of Science and Technology, Trondheim, Norway

[6]) Institute of the Urban Environment, Chinese Academy of Sciences, Xiamen, PR China

[7]) Institute of Social Ecology, University of Natural Resources and Life Sciences (BOKU), Vienna, Austria

Section involvement: A draft version of this guideline and good practice report was put out for public review and feedback via the ISIE homepage in the summer of 2023: https://is4ie.org/announcements/1703 The draft report was discussed several times in the Section Board meetings of the socio-economic metabolism (SEM) section of the International Society for Industrial Ecology (ISIE). In the summer of 2024, the Section Board endorsed the publication of this document.

Content:

List of acronyms and abbreviations:

DTTF:	Data transparency task force (of the ISIE)
FAIR:	Findable, Accessible, Interoperable, and Reusable
GHG:	Greenhouse gas
IE:	Industrial ecology
ISIE:	International Society for Industrial Ecology
JIE:	Journal of Industrial Ecology
LCA:	Life cycle assessment
MaTrace:	Material tracing, a specific type of dynamic MFA model
MFA:	Material flow analysis
MISO:	dynamic integrated model of material inputs, stocks and outputs
ODYM:	Open dynamic material systems model
RECC:	Resource efficiency-climate change mitigation model

1. Background and motivation

Data and data handling is at the core of industrial ecology (IE) and socio-metabolic research. With the growth and increasing complexity of IE research, partly driven by 'big data' and machine learning techniques, there is "a desire for better utilization of the accumulated data in more sophisticated analyses. This implies the need for greater transparency, accessibility, and reusability of IE data, paralleling the considerable momentum throughout the sciences" (Hertwich et al. 2018).

A core aspect of data transparency, accessibility, and reusability is the complete traceability of workflows and reproducibility of results, by which we mean

a system of procedures, documentation, and archiving for the work with data, including the identification of data sources, the processing of data, the assessment or model calculation steps, and the evaluation of assessment of model results.

There are four main reasons for installing such a system of procedures, documentation, and archiving:

- *Scientific rigor*: We must be able to link assessment and model results to raw data, to identify assumptions and proxy choices made, also after considerable time has passed since the research was conducted. This is a core requirement for validating the accuracy of findings through the scientific method, particularly amidst the prevalence of machine learning-generated (fake) outputs, and for detecting and rectifying errors in the data process.
- *Effective collaboration, update, and hand-over within teams, the IE community, and beyond*: In addition to making it easier to find and correct errors, a traceable data workflow enables effective collaboration by defining data interfaces and classifications, and by modularizing research so that work with data can be parallelized and experts from different disciplines (with specific data models and formats) can easily contribute. It also allows for quick updates of individual datasets (because outdated data can be easily identified), and large research projects can be handed off to new team members because of a well-documented and traceable data flow that is complete, easy to understand, and easy to update. The general scientific community (in IE and beyond) would also benefit. If data and code are easily available, other colleagues can directly build on your work and vice versa. This exchange is a precondition for accelerating the spreading of knowledge and increasing the quality and relevance of our work.
- *Demand by stakeholders:* Publicly funded sustainability science is increasingly expected to contribute to a cumulative knowledge base (Pauliuk 2020), which includes requirements to make research data accessible and to document research workflows.

- *Higher social trust and impact through higher quality research:* Unlike consulting, which often lacks transparency standards, scientific sustainability research earns the most trust by upholding rigorous quality standards. This includes fostering a culture of ongoing verification and error correction. In quantitative sustainability science, such a culture can only thrive if the data workflow is transparent and reproducible.

As a first step, "*the issue of data transparency was identified by the council of the International Society for Industrial Ecology (ISIE) as an important concern. The Society convened the Data Transparency Task Force (DTTF) in late 2016 to develop guidance on best practices and incentives for sharing IE research data and documenting research workflow.* […]" (Hertwich et al. 2018)

"The goal of the DTTF is to develop guidelines and incentives that encompass the whole research process, ranging from documenting input data and assumptions, to methodological aspects such as accessible software code, to providing access to generated output data." From (Hertwich et al. 2018).

Under its mandate, the DTTF did not address the question of traceability and reproducibility of the research workflow, and the data badges currently issued by JIE are only concerned with the data availability and reusability aspects of a research outcome.[1] Thus, to receive a (gold) data badge, the authors of an article have to specify the input data as well as publication of all research data outcomes of the analysis in a human and machine-readable format. How the authors derive the result data from the input data must be described in the article according to the state-of-the-art in each field, but transparent methodology and good traceability of results is not part of the assessment of the work for the data badge.

Good examples from the literature

Despite the lack of guidelines and standards, we increasingly see industrial ecology research articles, which make the full data pipeline publicly available.

Examples of high traceability of workflows and reproducibility of results range from articles that come with a traceable Excel workbook (Mayer et al. 2919; Haas et al. 2023), those that include the script used for the analysis (Wolfram et al. 2021), articles using R or Jupyter notebooks (e.g., (Vilaysouk et al. 2020; Boulay et al. 2021)), to full software suites including the model packaged as a standalone software package (e.g., (Kuczenski et al. 2022)). A number of gold-gold badge articles in the Journal of Industrial Ecology not only make their input and result data available, but also provide

[1] https://jie.yale.edu/data-openness-badges

the analysis and detailed documentation, e.g., (Mayer et al. 2019), (Harpprecht et al. 2021; van der Meide et al. 2022; Steubing et al. 2022).

Clearly, parts of the Industrial Ecology community already explore options for a fully transparent and open research workflow. The International Society for Industrial Ecology (ISIE) and its topical sections should use this momentum to provide incentives, good and best practice examples, and guidelines for complete traceability of workflows and reproducibility of results.

Expanding on existing guidelines

In 2021, the SEM (socio-economic metabolism) section of the ISIE (International Society for Industrial Ecology) issued guidelines for research involving the method of material flow analysis (MFA) (SEM Board 2021).

In 'Guideline VI: Data provenance and traceability' of the guideline document (SEM Board 2021), the following is written:

> "The data used for quantitative research undergo many transformations. First, they are extracted from their sources (sometimes manually, by reading and re-typing them!), then revised, amended, reformatted/reshaped, combined, stored in different formats, and finally used for calculating model results or indicators or for plotting them or presenting them in some other form. Model and calculation results are processed into scientific output, typically, by plotting aggregated or selected results or by reporting central numerical results in scientific reports and papers.
>
> *Documenting data flows across different tools requires special attention.* [highlighted by the authors of this work] […]
>
> While documentation procedures are project-specific and have to be adapted to each tool chain, only general hints are given here. In particular, MFA researchers and practitioners should pay attention to the following steps:
>
> - *Data sourcing:* Document the exact locations or identifiers (in documents or databases) of all data used, plus the dataset's version number (if any).
> - *Data processing:* Document the entire research flow: all modification and processing of the data into numerical results, both the different data treatment steps and the different tools and interfaces used.
> - *Data visualization and reporting:* Document exactly how the data were aggregated and visualized for a paper or report. Make sure that each number, figure, and table in your paper and the data therein can be traced back to the very model and data versions that were used to create those visuals in the first place.

- *Data archiving:* Follow the best practice set by the journals you publish in[2] and consider uploading datasets to public archives (e.g., Zenodo).

These steps are crucial to ensuring exact reproducibility of results, checking correctness, updating (modular) calculations, and supporting the re-use of data. They also help establish informational independence, because if two MFAs are actually based on the same underlying data, they are not providing independent information."

In the meantime, a number of developments for complete traceability of workflows and reproducibility of results have occurred in the research community for socio-economic metabolism (SEM).

Many colleagues are working on developing the state of the art of our research towards better traceability of results and data, which increases the quality of the work, helps with the reuse of data and results, and facilitates updating, correction, teamwork, and handover of work to other colleagues.

In this document, we want to add detail to the general recommendations for data provenance and traceability given in (SEM Board 2021), compile a number of working examples, and give an outlook on current trends and future development options.

These specific examples drawn from SEM research are intended to be complementary to the many other excellent resources on reproducible research more generally, such as The Turing Way.[3]

[2] e.g., https://jie.yale.edu/data-openness-badges

[3] https://the-turing-way.netlify.app/index.html

2. Five steps for complete traceability of workflows and reproducibility of results

In our research, each workflow is unique, however, a standard pattern from raw data to processed data and further to model-based results is part of every quantitative research project. For this standard pattern, we identified the following general procedural steps for complete traceability of workflows and reproducibility of results (Box 1, illustrated in Fig. 1).

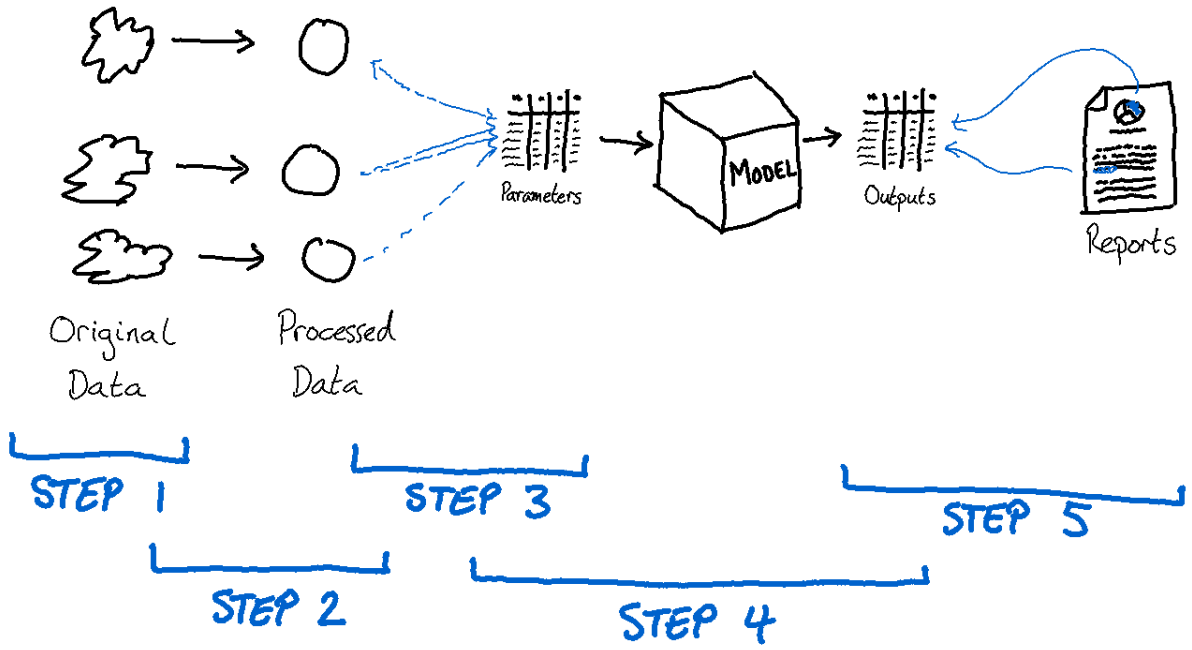

Fig. 1. Illustration of the five steps for complete traceability of workflows and reproducibility of results, drawn by our co-author Rick Lupton. Model output often is the new original data for subsequent analysis, and it needs to be archived with sufficient reproducibility and traceability information.

The list in Box 1 explains these five steps and gives guidelines for their implementation. It builds on published concepts for modelling in industrial ecology (Pauliuk et al. 2015) in that it uses the same basic steps and adds the specific routines that need to be followed[4]. Details are provided in the following subsections.

[4] In machine learning and data science, steps 1-3 are commonly referred to as the ETL (extract, transform, load) pipeline.

Box 1: Five Step Guideline for Ensuring Data Processing and Results Reproducibility

Each step explains why it is important and gives examples of how it can be achieved.

1. **Ensure the traceability of the original data used**, to ensure future access.
    a. Link data to their original sources, using DOIs or other identifiers
    b. Archive downloaded data and keep an inventory of raw data files
    c. For databases: Write a query script, document data version/access date

2. **Document how processed data was derived from original data,** to trace each single data point to its respective original sources and assumption.
    a. For small data, this can happen with equations etc. in a spreadsheet.
    b. For larger data, the processing of raw data should be documented in scripts that use standardized data handling routines.
    c. For a broad mix of data sources, data processes (section 3.5) can be used.

3. **Keep track of how processed data are combined into datasets that form the model parameters,** so the origin and processing of all data in the model input parameters can be understood.
    a. Keep a record of all modifications to the model parameters, i.e., version-track the model parameters when changing them.
    b. Keep an inventory and version list of your entire model/assessment database, ideally in a traceable online repository such as GitHub.
    c. Databases should see regular backups, and data should ideally not be overwritten but kept.

4. **Document relevant calculations and model runs**, so they can be reproduced.
    a. List tool and package versions used and describe computational environments.
    b. Document versions of model components and input data for each run.
    c. Keep track of the results of each relevant run.

5. **Link final outputs to the calculations that produced them,** so figures and data in reports and papers are traceable**.**
    a. Link results in a report to a model run ID (can again be linked to database ID, model version ID, and computational environment ID)
    b. Document exactly how numbers in tables and plots are linked to/derived from model results.
    c. Document exactly how numbers in the text are linked to/derived from model results.

2.1. Details for the five steps of data processing and results reproducibility

In the following section, the five steps suggested above are described and discussed in more detail. Section 3 provides a number of good practice examples of how these steps can be implemented in research practice, ranging from spreadsheet workbooks to comprehensive data processes for specific modeling environments.

Step (1) Ensure the traceability of the original data used

More and more often, IE researchers build their assessments and models on specific archived versions of data sets published in an established data repository or along with scientific publications, such as *Zenodo*. This allows them to (automatically) download the data and to specify the DOI (or other identifiers) in their analysis scripts. This way, a traceable link between input data and their use in own research is established.

Still, data often is only available through generic API access, via proprietary databases, or need to be read and parsed offline. Examples include the digitization of historical statistical data, which often happens manually, from scanned books or pdf reports. In many cases, these raw data cannot be shared with others. Confidential raw data cannot be shared either. API access can lead to another reproducibility problem. Many data providers, including most statistical agencies, update the numbers of their database under the same API access point, without allowing users to trace or switch back to previous versions. Running the same code with these data might lead to different results, depending on when it is run. Previous versions of the data are often no longer available at all. Moreover, when the license of the API-download does not allow for republishing them, results can only be reproduced locally. Therefore, in the absence of permanent URLs or POI (e.g., DOI) or under restrictive licenses, it is advisable to store a local copy of the downloaded data for documentation via self-archiving of downloaded data files in a *raw data* folder (with exact source and time stamp). The source files (pdfs, spreadsheets, csv files, images, etc.) are archived locally (e.g., in a folder tree by parameter, sector, material, or country) so that each data source has a unique relative path in the archive and can be accessed at any time.

Step (2) Document how processed data was derived from original data

Ideally, the whole data process runs through a defined, version-managed, and well-documented data pipeline to meet the goals outlined above.

For example, each data file can be accompanied by a log file that contains the necessary records to trace each single data point to its respective data process and original sources and assumptions.

Basic reproducibility can be achieved through thorough documentation of the steps to be followed, but it is usually better to make use of tools define precisely and automate the steps. These can be plain scripts (e.g., in Python or R, or shell scripts) or can make use of task runners such as (Snake)Make, Doit,[5] or Targets.[6]

Code notebooks such as Jupyter notebooks,[7] Quarto,[8] Pluto,[9] or R Markdown documents[10] are common ways of combining calculations and their documentation and are increasingly used by IE researchers. They can be integrated into reproducible workflows using packages such as rrtools,[11] see the examples below. These tools typically contain model calculations and thus cover steps 2-4 of the general guidelines in Box 1. They can also be used for step 5 (make final results in reports and papers traceable), see the examples below.

Semantic data technologies such as RDF are an alternative possible solution to the issue of keeping track of data provenance (Germano et al. 2021).

Step (3) Keep track of how processed data are combined into datasets that form the model parameters

A project or model input database should be divided into different data types and data files (for larger projects), depending on the origin and the use of the data in the model/assessment. To keep an overview of the project's database, an inventory and version list of your entire model/assessment database is needed, including a change log that records all additions and changes to the database.

For example, in a spreadsheet workbook, identify smaller sets of processed data as well as input data/parameters clearly in a separate sheet, and keep copies of alternative versions. For a larger database, keep a log file (either project-wide or data file-specific) that contains a record of all modifications of the model parameters, so that one is able to trace each single data point in each model input parameter to its respective original sources and assumptions. For larger, data-heavy projects, a structured log file format should be used to allow queries against the log entries. Depending on the use case, the logging might also need to be done with a database, particularly when multiple value changes of specific data entries must be tracked.

Where possible, prefer simple and text-based formats for storing data and keeping track of changes. It is much easier to use tools like git to track changes to a CSV file than tracing changes that are hidden in a large Excel workbook. Of course,

[5] https://pydoit.org/
[6] https://books.ropensci.org/targets/
[7] https://jupyter.org/
[8] https://quarto.org/
[9] https://github.com/fonsp/Pluto.jl
[10] https://rmarkdown.rstudio.com/
[11] https://github.com/benmarwick/rrtools

sometimes the benefits of organizing data into one workbook may outweigh the benefits of easier change-tracking. Especially, since Excel is very useful also when collaborating, as many people are used to it and can therefore work with such data. Examples for both types of approach are given below.

Step (4) Document relevant calculations and model runs

Quantitative assessments rely on calculations to derive indicators. To trace these indicators back to their data and model origins, it is crucial to establish traceability by thoroughly documenting the model calculation process. Typically, each model run will create a log file, where the version numbers of all model components and all input data are documented, so that the model/assessment results are then linked to model and data versions by this log entry. As an alternative, the procedure of obtaining results may be documented along with the data, for example, in LCA: "These results were obtained with exactly this foreground process data file (link to exported foreground data file), this background database version, this software version, and on this computer system." This requirement for model run tracing goes hand in hand with the recommendation for FAIR (Findable, Accessible, Interoperable and Reusable) research software, as lined out by Barker et al (2022).

Most research code and analyses evolve incrementally, and during this process, it is unknown which version will be the final version. To keep an overview of different iterations and ensure reproducibility of previous results, we recommend using some form of version control (i.e., git) to document and preserve each step of development. Some of the standard tools for version control and scientific repositories are also well connected. For example, it is possible to link a GitHub repository to a Zenodo archive, which then automatically archives each released version of the model/code.[12] Similar possibilities exist for *figshare*, *osf*, and other archives.

Step (5) Link final outputs to the calculations that produced them

For all display items (figures and tables) as well as all quantitative statements in the publication, an explanation should be provided on where exactly these numbers come from (e.g., exactly which model run for data shown in figures and tables), or how exactly the number mentioned in the text was obtained. (e.g., "The *xy* Mt of CO_2-eq stated in the abstract are calculated as the average of column *a* on sheet *b* in the *c* result workbook archived under URL *d*, rounded to the nearest Mt.") Such information could be compiled in the supplementary material.

LaTeX typesetters like overleaf[13] allow for importing numbers and tables into text documents via plain text import or hyperlinks, allowing for automatic updating of

[12] https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content

[13] https://en.wikipedia.org/wiki/Overleaf

manuscripts as new data become available. For example, a Python or R script can generate a .tex file coding a table with numeric results in it, which is then inserted into the final text during each recompilation of the LaTeX document from its source code. Next to convenient updating, the link between text document and model code via a .tex file allows for tracing the information flow from the model to the paper.

Notebooks such as Jupyter, and tools like Quarto[14] also help with this by allowing code that generates a table to be embedded directly in a document, with the results automatically embedded into the final LaTeX/PDF/HTML output.

[14] https://quarto.org/

3. Good Practice Examples for Workflow Traceability and Reproducibility of Results in Industrial Ecology Research

Below, we provide a detailed description of several approaches for data provenance and traceability. The examples below were provided by members of our community, and we list them here as good examples that show what is currently done and what development options exist. This list of examples is not exhaustive, and we are happy to include other approaches in future updates of this document.

3.1. Spreadsheet calculations

"In principle, an Excel workbook (stored as .xlsx file) contains and documents (if well-structured and annotated) all the above-mentioned steps in an open file format, starting from data queries and ending with internal plots and summary tables. In practice, however, such workbooks quickly get messy, do not offer enough computational capabilities, and cannot handle the given data volume. For many research applications, different tools and combinations of tools need to be used anyhow." (SEM Board 2021)

Although the limitations of spreadsheet software, such as *LibreOffice* or *MS Excel* for IE research become quickly apparent, they are widely used for smaller projects or for auxiliary calculations with specific data as part of larger projects because of their practicality and integration of human and machine readability. In Box 2, we propose how the general steps for complete traceability of workflows and reproducibility of results can be implemented for spreadsheet calculations.

Box 2: Proposal for the implementation of the five general steps for complete traceability of workflows and reproducibility of results for spreadsheet calculators (e.g., *LibreOffice* or *MS Excel*)

1. **Ensure the traceability of the original data used**

   Each spreadsheet workbook should have a separate reference sheet, where the different data sources (URL + access date, DOI, etc.) are listed. The calculations in the workbook refer (via text or comments) to specific data from these references.

2. **Document how processed data was derived from original data**

   A copy of the original data should be created (e.g., on separate sheets), and all data processing and modification shall be coded (using calculations/equations and color codes) or clearly explained otherwise, so that all steps between raw data and processed/final data are *coded and documented* in the workbook. It is crucial to avoid typing numbers into cells manually when transferring them from other cells. Instead, cells should be linked directly, via equations. Data links across workbooks, however, shall be avoided, as they break easily when files are moved.

3. **Keep track of how processed data are combined into datasets that form the model parameters**

   Each relevant version of the workbook has a unique filename and a version number, ideally accompanied by a log sheet, where researchers log and time stamp their activities.

4. **Document relevant calculations and model runs**

   For self-contained spreadsheets, this step is identical with step 3, as the results are contained in the same document as the input data and calculations.

5. **Link final outputs to the calculations that produced them**

   Model result figures and tables should be contained in the same workbook so that the links between result data and display items are documented. For all quantitative statements in the publication, an explanation should be provided on how exactly this number was obtained.

The recommendations above also apply when spreadsheet calculations are used as part of larger projects, e.g., to document ancillary calculations or when evaluating model results.

One example for a spreadsheet-based workflow is the macro-level circular economy assessment presented by Mayer et al. (2019), who document and provide their complete workflow as excel file. The workbooks links official data inputs through the entire calculation via excel links. The work also contains a pdf documentation with all definitions and further information and assumptions. This article was the first to get the gold transparency badge from the Journal of Industrial Ecology.

3.2. Example for spreadsheet data: Material flow analysis of critical raw materials by Peng Wang and colleagues at the Institute of Urban Environment, Chinese Academy of Sciences

Critical raw materials are important for global low-carbon and sustainable transition, but there is a notable gap regarding their supply chains transparency that prohibit robust decision making. Material flow analysis is an important tool for tracing the material flows of various raw materials. However, it often malfunctions when it comes to certain critical raw material. This is particular due to the lack of data transparency along supply chains and the existence of various unregistered mining and trade activities. Here, this chapter shows how one can incorporate the guidelines and procedures with a recent study of global rare earth flows (Chen et al., 2024), as one example of how to collect original data, process it, calculate, and finally interpret the results of MFA studies for critical minerals. The following five steps are taken:

Step (1) Collecting dispersed data under a common system definition

In this step, a systematic material flow framework was developed, as shown in Fig. 2, which clearly identifies the necessary production, consumption, loss, and trade data. Multiple data sources, such as national statistical yearbooks, published books, academic literature, and industry reports are typical data sources. Notably, trade data from UN Comtrade is limited to one general six-digit Harmonized Commodity Description and Coding Systems (HS) code, and is often not suitable for the study of critical minerals. Accordingly, a higher-resolution (eight/ten-digit HS codes) trade data of critical minerals from national customs records was obtained.

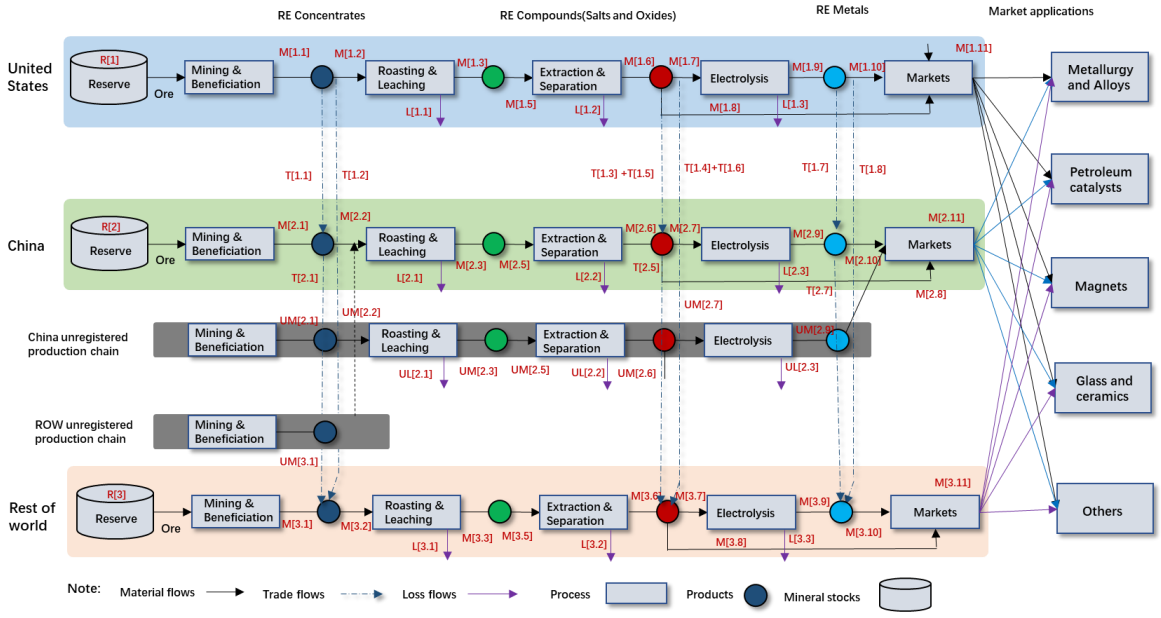

Fig. 2. Material flow quantification framework of rare earth flows across stage and national borders, showing the data sources directly in the MFA system definition. Image source: Chen et al. (2024).

Step (2) Documenting how processed data was derived from original data

In this step, this study provides a detailed table (see Fig. 3) to show how all origin data are quantified and converted to metallic equivalents, using mass balance principles based on various publicly available statistics and reports.

Symbol	Name	Comments
R[1]	Rare earth reserve	USGS
M[1.1]	Mine production	USGS mineral yearbook from 2000 to 2018
T[1.1]	RE concentrates net export from U.S. to China	From trade data based on commodity codes in Table S2
T[1.2]	RE concentrates net export from U.S. to RoW	M[1.1] -T[1.1](mass balance)
M[1.2]	Mine input to leaching	M[1.3]/ 85% RE recovery rate of Roasting and leaching: 85%[14]
L[1.1]	Leaching losses	M[1.2]- M[1.3] (mass balance)
M[1.3]	RE salts production	No refinery production capacities in U.S.[15]
M[1.5]	RE salts inputs to extraction	M[1.6]/96% RE recovery rate of extraction: 96%[16]
L[1.2]	Extraction losses	M[1.6]- M[1.5](mass balance)
M[1.6]	RE oxides production	No refinery production capacities in U.S.[15]
T[1.3]	RE salts net export from U.S. to China	From trade data based on commodity codes in Table S2
T[1.4]	RE salts net export from U.S. to RoW	From trade data based on commodity codes in Table S2
T[1.5]	Oxides net export from U.S. to China	From trade data based on commodity codes in Table S2
T[1.6]	Oxides net export from U.S. to RoW	From trade data based on commodity codes in Table S2
T[1.7]	Metals net export from U.S. to China	From trade data based on commodity codes in Table S2
T[1.8]	Metals net export from U.S. to RoW	From trade data based on commodity codes in Table S2
M[1.7]	Oxides input to Metalmaking	M[1.9]/ 97% Metal recovery rate of electrolysis: 97%[17]
L[1.1]	Metalmaking losses	M[1.2]- M[1.3](mass balance)
M[1.9]	RE metal production	T[1.8] (mass balance)
M[1.8]	RE oxides domestic consumption	M[1.6]-T[1.5]- T[1.3]- T[1.4]-T[1.6]-M[1.7]

Fig. 3. One example for detailed description of equations, conversion factors, and data sources to support material flow analysis. Image source: Chen et al. (2024).

Step (3) Documenting relevant calculations and model runs

In this step, all calculations are based on the mass balance principle, which means the total input is equal to the total output for each process in the system Importantly, special attention should be given to unregistered production and trade routes of critical minerals. The calculation of registered production is mainly production-driven, which starts with the mining production estimate and continues with quantifying the rest of the flows based on the mass balance principle. A trade-driven approach (based on the gap from one the trade record of one country and the other country's trade data) is followed to quantify the flows along this route.

Step (4) Result visualization and validation

In this step, the results of one particular year (e.g., 2020) are derived (Fig. 4). After that, dynamic year-by-year results from 2000 to 2022 can be derived with an analogue procedure. Validation is achieved by comparing selected aggregated results with the results reported by other studies.

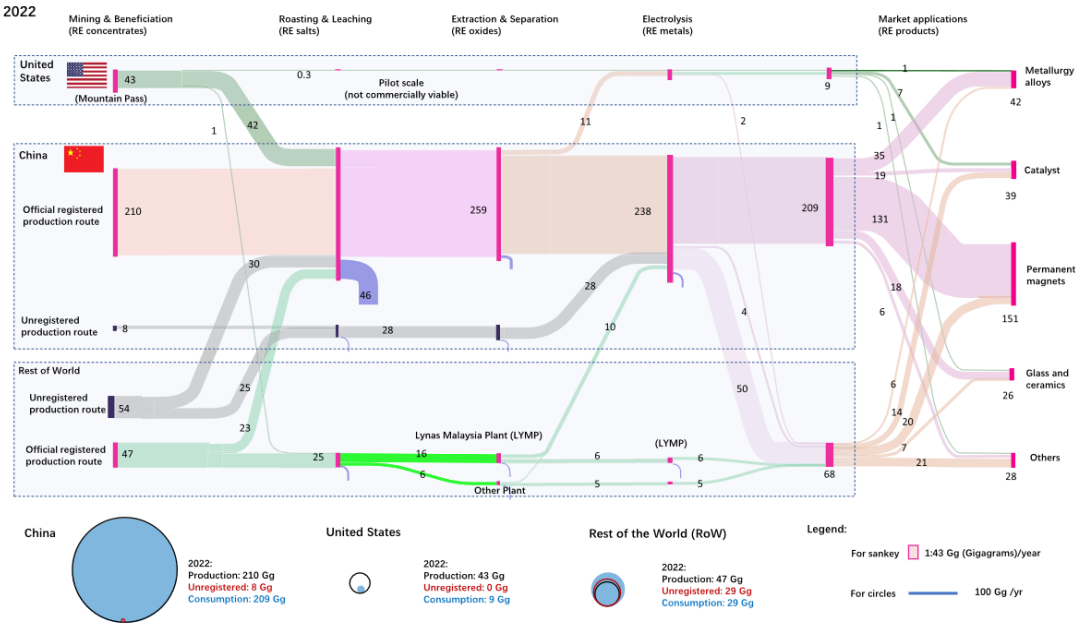

Fig. 4. Result visualization as a Sankey diagram (for each year from 2000 to 2022). Image source: Chen et al. (2024).

Step (5) Linking final outputs to the calculations that produced them

In this step, a traceable Excel workbook containing origin data, processed data, the calculation equation, and results was provided (Fig. 5). All of the data and results are open access in the data file deposited at https://zenodo.org/records/10396895.

	Symbol	Name	Value (2022, kt)	Value (Total, kt)	
U.S. Production route	R[1]	Rare earth reserve	1400	1400	Production database
	M[1.1]	Mine production	43	195.4	Production database
	T[1.1]	RE concentrates net export from U.S. to China	41.7	163	Production database
	T[1.2]	RE concentrates net export from U.S. to RoW	1.0	31.7	M[1.1] -T[1.1], mass balance
	M[1.2]	Mine input to leaching	0.30	0.45	no capacity
	L[1.1]	Leaching losses	0.05	0.07	no capacity
	M[1.3]	RE salts production	0.26	0.38	no capacity
	M[1.5]	RE salts inputs to extraction	0.26	0.38	no capacity
	M[1.6]	RE oxides production	0.26	0.38	no capacity
	M[1.7]	RE oxides to metal productoin	0.25	0.37	no capacity
	L[1.2]	Extraction losses	0	0	RE recovery rate: 96%
	T[1.5]	Compounds(salts &oxide) net export from U.S. to China	-7.7	-260	Assume no export from us to China (cause the data results are negative number)
	T[1.6]	Compounds(salts &oxide) net export from U.S. to RoW	-7.0	-28	Trade database
	T[1.7]	Metals net export from U.S. to China	-0.5	-50.1	Trade database
	T[1.8]	Metals net export from U.S. to RoW	0.1	58	Trade database
	M[1.9]	RE metal production	0.25	0.37	T[1.8]
	M[1.7]	Oxides input to Metalmaking	0.3	0.4	M[1.9]/Metal recovery rate: 97%
	M[1.8]	RE oxides domestic consumption	14.7	288.1	M[1.6]-T[1.5]-T[1.6]-M[1.7]
	M[1.10]	RE metal domestic consumption	0.6	-7.3	M[1.9]- T[1.7]- T[1.8]

Fig. 5. Result visualization for the results of Chen et al. (2024). Image source: Chen et al. (2024).

Core reference:

Chen, W.-Q., Eckelman, M. J., Sprecher, B., Chen, W., & Wang, P. (2024). Interdependence in rare earth element supply between China and the United States helps stabilize global supply chains. One Earth, 7(2), 242–252. https://doi.org/10.1016/j.oneear.2024.01.011

3.3. Data workflow based around Python at the University of Bath (Rick Lupton)

Our research group uses tools from the Python ecosystem to carry out analysis. Here we give some examples of how we use these tools. This is a journey – it can be difficult to get everything right alongside the demands of refining research questions, collecting data, modelling, presenting findings clearly, and so on. As we gain more experience with what works, we are working to embed best practice in our training of new researchers and across our projects.

Overall approach to working with reproducible research repositories

In our research we structure work around version-controlled git repositories. The first choice to make is how to structure these – a repository may be linked to a dataset, a model, or a particular analysis. We aim to break up elements of work into chunks according to how specialized/reusable they are. Each chunk then naturally focuses on a different one of the five steps discussed above.

For example, a modelling project might involve:

1. Collection & cleaning of data: this is the most generic/reusable part (steps 1, 2)
2. Modelling calculations (e.g. MFA reconciliation): this brings in more assumptions about the model structure to be used, so is less generic than the original data. It results in potentially many detailed outputs, more than would be shown in a specific report. It therefore sits at an intermediate stage (steps 3, 4).
3. Specific data and figures for a specific document, derived from the model. (step 5).

Sometimes the scope of the work is small enough that there is no need to split these into separate repositories. But it can be useful, since it allows other researchers who are interested to build on the original data to do so more easily, without having to figure out how the data integrates into a model or figures they are not interested in reusing. It also has the benefit of checking that good practice has been followed in documenting and clearly structuring each stage, since otherwise it is difficult to disentangle the work into separate repositories. Tools such as Datalad, described below, make working with multiple repositories in this way easier.

One example of this approach in practice is our modelling for a report on UK steel flows (Allwood et al, 2019), where data, modelling and figures were arranged across three repositories:

- Collection and cleaning of steel trade data: https://github.com/ricklupton/uk-steel-trade (Lupton & Serrenho, 2019a)

- Modelling: https://github.com/ricklupton/uk-steel-model (Lupton & Serrenho, 2019b)

- Report and figures: https://github.com/ricklupton/steel-arising-report (Lupton & Serrenho, 2019c)

For another study comparing datasets on global petrochemicals production and emissions (Malkowska et al, 2024), we followed a similar approach where each dataset is first converted (in self-contained repositories) into a common RDF-based structure (Germano et al, 2021). Each of these converted datasets is potentially useful as a building block for further unrelated analysis, so it is useful to have them as separate repositories. For example, UNFCCC emissions data is converted in https://github.com/probs-lab/unfccc-data.

Python computational environment (step 4)

To describe the Python computational environment (the version of Python and all the tools and packages needed to run the code), we used Anaconda environments based on conda-forge (https://the-turing-way.netlify.app/reproducible-research/renv/renv-package.html, https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html), or Poetry (https://python-poetry.org/). Anaconda is more powerful and allows a wider range of scientific packages to be installed across operating systems in a consistent way, but Poetry integrates better with the basic Python tools and is simpler if it does all that is needed. Pipenv is an alternative tool we have used in the past for the same purpose as Poetry.

- Example: https://github.com/ricklupton/uk-worldsteel-statistics

- Example: https://github.com/probs-lab/unfccc-data/blob/3082428645ecd293270bd41b78011cb3e7b9be85/environment.yml

Reproducible data processing and analysis steps (steps 2–5)

Different tools can help with documenting the steps needed to reproduce an analysis

- Make is a classic tool to describe computational steps and the dependencies between them (https://github.com/ricklupton/uk-steel-model/blob/v1.0.0/Makefile)

- Wrapping the full list of steps in a single build script makes it explicit exactly what someone needs to do to rerun the analysis (https://github.com/ricklupton/uk-steel-model/blob/v1.0.0/build.sh)

- Pydoit (https://pydoit.org/) is an alternative to Make based on Python which makes it easier to describe more complex tasks, such as "run this command on every country's data. It saves the result with the same filename in a new folder, but only if

the input data has changed more recently than the existing outputs" (https://github.com/probs-lab/probs-ontology/blob/v1.5.2/dodo.py).

Storing and retrieving specific versions of large data files (step 1, step 5)

Sometimes the data files themselves are too large in size or too numerous to want to store them in the same place as the analysis code and documentation. For example, we use Github extensively to store and share code and documentation, but git and Github work less well when storing large quantities of data, especially when there are different versions of the data files to track over time (e.g. as new input data is published, or model outputs are updated). We use a tool called Datalad (https://www.datalad.org/) to help keep track of versions of data files. This allows for:

- Fetching only the specific data files you want, to avoid downloading lots of data you don't need
- Tracking and switching between versions of large data files easily.
- Linking output data from one analysis as input data to another, automatically retrieving the previous outputs when needing to run the new analysis – this makes it easier to work with multiple repositories, as described above
- Storing data in a separate location from your code and documentation; the data can be stored in a private location if necessary, allowing as much reproducibility as possible even when some input data is confidential.

These features come with some costs – there is more to understand and learn, which may not be worth it in every case. But the ability to set up a copy of an analysis on a new researcher's computer and automatically retrieve the exact versions of the necessary input data in a single command is useful.

For example, our UNFCCC-data repository[15] describes how copies of the correct version of all the input files needed can be retrieved by running a single command.

[15] https://github.com/probs-lab/unfccc-data/blob/3082428645ecd293270bd41b78011cb3e7b9be85/DEVELOPING.md#data-access

Box 3: Proposal for the implementation of the five general steps for data workflows based around Python, by Rick Lupton

1. **Ensure the traceability of the original data used**
    a. Where possible, link data to their original sources in long-term archives using DOIs. Use tools such as Datalad to automate fetching the correct version of data, and to fetch only the specific files needed (example: https://github.com/probs-lab/unfccc-data)
    b. Alternatively, archive downloaded original data files within the project in a "raw data" folder which is clearly separate from processed data and model outputs.
2. **Document how processed data was derived from original data**
    a. Use Python scripts which read the original data and write a new version to a separate folder. Use task runners like "doit" to help to make sure all necessary processing steps have been re-run when input data or scripts change, but avoid unnecessary re-running when nothing has changed.
3. **Keep track of how processed data are combined into datasets that form the model parameters**
    a. We tend to use a mix of input data files and modelling code, which are stored in a version-controlled git repository. Each snapshot of the repository therefore represents all the inputs to the model. However, in reflecting on how our approach fits with these five steps, this may be an area that could be improved, since the full set of input parameters and assumptions may be scattered across several files and scripts and not easy to see in one place.
    b. Input data must be linked back to its original/processed source and/or assumptions, in the form of code comments, additional columns in spreadsheets, etc.
4. **Document relevant calculations and model runs**
    a. To recreate the computational environment, we typically use Anaconda "environment.yml" files or the Poetry Python environment manager to specify exactly the version of each package or tool needed to run the analysis. To make sure this really works, testing between team members on different computers is important, attempting to follow the process and checking that nothing is missing from the environment or out of date.
    b. By working in a git repository, when data files are not too large, model results can be committed to the repository alongside the input data and code. In this way, the model code, input data, and resulting outputs are clearly tied together, and different versions of outputs can be easily compared.

Box 3 ctd.:

4. **Document relevant calculations and model runs (ctd.)**

    c. When data files are too large to store them directly in git, Datalad helps to keep the same workflow while allowing the actual data to be stored elsewhere (e.g., on a network drive)

5. **Link final outputs to the calculations that produced them**

    c. To document exactly how numbers in tables and plots are linked to/derived from model results, we typically produce plots using scripts or Jupyter notebooks which can be easily re-run. The code of these scripts traces which model outputs are used to create them.

    d. Although it would be nice to do everything using tools such as Quarto with embedded inline code to create fully reproducible documents, currently we work in a more pragmatic way where final reports are often created manually. A complete README file for the notebooks or scripts which produce the figures then sets out which figure is produced by each notebook.

3.4. Data workflow based around R at the Social Metabolism and Impacts research group at PIK, Germany

In our FutureLab Social Metabolism and Impacts at the Potsdam Institute for Climate Impact Research, we have been working for several years to implement reproducible workflows for all our projects and publications. We strive for at least full computational reproducibility, which means that others can obtain the same results using the same data and code, because we believe that a well-designed reproducible research pipeline helps the researchers themselves, their collaborators, and the scientific project as a whole. However, we also recognize that it is not easy to find a one-size-fits-all approach to computational reproducibility in a highly inter-disciplinary research group, where people conduct very different kinds of research. We have found that such a process can only be introduced gradually and requires a lot of active discussion and mutual learning. This is especially true given that the skills needed, such as data management, software development, and visualization, are becoming increasingly important in research practice, but are often not taught at universities.

One-click reproducibility

We aim for what we call one-click-reproducibility. This means we want to go from from raw data to the finished report or scientific article with a single click (or command). For this, it is important to avoid any manual data manipulation (e.g., in EXCEL) or figure creation (e.g., in Inkscape). In our group, we use literate programming with R and RMarkdown for analysis and documentation and Gitlab for publication. RMarkdown allows generating reproducible documents (MS Word, pdf, html, etc.) by weaving together executable code and textual elements based on plain text markdown. The use of plain text (e.g., Markdown or LaTeX) has the additional advantage of allowing for version control (e.g., using git) for manuscript writing.

Overall, in our view, a fully reproducible research pipeline requires publication of all data, software code, the R environment and the wider computational environment.

We use the *rrtools* package for creating a reproducible research compendium. The *rrtools* package takes care of setting up your git repository, a project folder structure, a README file describing the project, and required license files. It relies on the *renv* package to make your R environment reproducible and allows setting up a *docker* container to preserve your entire computational environment.

For small empirical projects, the analysis code is included directly in the RMarkdown file or in individual scripts stored in the compendium. If the code base developed is reused across publications or projects, we encourage the creation of a dedicated R package for that code. R package development is supported by a number of tools that make code development, documentation, and testing much easier than simply writing

individual scripts. For projects that require code with long runtimes, we recommend using the *targets* package, which keeps track of internal code dependencies and speeds up execution by rerunning only code affected by recent changes.

The following list contains several example repositories for recent publications, where we applied most or all of these principles:

Pichler, P.-P., Jaccard, I. S., Weisz, U. & Weisz, H. International comparison of health care carbon footprints. Environ. Res. Lett. 14, 064004 (2019).

Jaccard, I. S., Pichler, P.-P., Többen, J. & Weisz, H. The energy and carbon inequality corridor for a 1.5 °C compatible and just Europe. Environ. Res. Lett. 16, 064082 (2021).

Belmin, C., Hoffmann, R., Elkasabi, M. & Pichler, P.-P. LivWell: a sub-national Dataset on the Living Conditions of Women and their Well-being for 52 Countries. Sci Data 9, 719 (2022).

Belmin, C., Hoffmann, R., Pichler, P.-P. & Weisz, H. Fertility transition powered by women's access to electricity and modern cooking fuels. Nat Sustain 5, 245–253 (2022).

Box 4: Example for the implementation of complete traceability of workflows and reproducibility of results using R/Quarto

1. **Ensure the traceability of the original data used**

Place data in the raw data folder of your project and document each file in a text file or a preprocessing script (incl. URL + access date, DOI, etc.). Download results of API calls (e.g., World Bank) into raw data folder if data version can't be specified.

2. **Document how processed data was derived from original data**

Use a clearly named preprocessing script for all preliminary transformations on your raw data, and (optionally) store the result in a separate folder for derived data. It is often beneficial to store preprocessed data and avoid loading raw data outside of the preprocessing to ensure that data that is loaded multiple times throughout the project is always identical.

3. **Keep track of how processed data are combined into datasets that form the model parameters**

Use version control (git) with verbose commit messages. This makes sure code and parameter setup can be reproduced later.

4. **Document relevant calculations and model runs**

In addition to versioning your own code, you should document your computational environment. This can be done in several ways. The simplest is to store the version of R and the version of any packages you use, either in a file or using a package like *renv* (https://rstudio.github.io/renv/). More thoroughly, create a *docker* image of the entire computational environment, including a snapshot of the operating system and all the libraries and packages needed to run the code).

5. **Link final outputs to the calculations that produced them**

Create all figures/tables and your manuscript using Quarto/Rmarkdown documents, this will ensure that these elements all update automatically with code/data changes. Do not hardcode result numbers into your manuscript text, use inline code instead.

3.5. Data workflow by Simon Schulte, Industrial Ecology Freiburg, based on R

My R-based workflow resembles the one from PIK described in chapter 3.4 by building on the same principle of "one-click reproducibility". It undergoes constant development, as limitations emerge to my current solutions and I discover new tools, hence, I am only sharing a snapshot of my current setup here.

I use *renv* to build a virtual environment with all R package dependencies tracked by version number. For subsequent code users, *renv* installs the required versions of all packages. For each project, I have one config file (through the *configr* R-package), where a version number has to be specified besides all relevant global settings of variables and paths. Each script reads and writes to a designated subfolder, only storing intermediate results for that version number. Each time, a copy of the script itself is saved in the same intermediate results folder so that the intermediate results can be traced both by its version number and the file that created the output.

Experiences with literate programming:

During my doctoral studies, I experimented with literate programming tools (Rmarkdown and Quarto) to write reports, scientific publications and presentations embedding R code. Literate programming tools, such as Quarto, can create dynamic documents that can be recreated when the input data or calculations change. In Quarto, this is achieved by creating figures, tables, and numbers within the document itself using R, Python, or Julia code chunks. Thus, instead of writing "[...] accounts for 20% of global GHG emissions," you write, e.g., "[...] accounts for `r x[1] / sum(x)` of global GHG emissions" if the variable x stores a vector of global GHG emissions differentiated by emission sources. Each time x is updated, the document can be recreated, and the numbers in the text are updated accordingly.

For very small projects, all calculations can be done within the same quarto manuscript. For larger projects, it is advisable to do computationally expensive calculations beforehand in one (or several) R/Python/Julia scripts and only load the final results into the Quarto script. Each time the model is re-run and new results are created, the Quarto script can be executed, and all numbers, figures, and tables are automatically updated.

Quarto has the advantage that

- figures, tables and numbers that appear in the manuscript are fully traceable
- it saves time updating the manuscript when data changes
- it is less error-prone because it prevents that one simply forgets about updating numbers/figures/tables
- it supports R, Python and Julia code chunks
- it supports multiple output formats: PDF or LaTeX for scientific publications, HTML for presentations or websites, MS Word, and MS PowerPoint. Hence, you can create a presentation or dashboard from your paper without much additional effort.
- it allows for version control with git

Despite these advantages, collaborating with co-authors is a yet unresolved obstacle. As of today, there is no tool that I am aware of that makes collaborative editing and commenting on manuscripts as convenient as Google Docs or Overleaf. This applies in particular to those collaborators who are not familiar with quarto/R and are mainly interested in editing the written text rather than the underlying calculations. There is an R-package called *trackdown* that automates the upload of quarto files as GoogleDoc, which then can be shared among collaborators and edited/commented. However, compared to Overleaf, there is no side-by-side preview of the final PDF version of the manuscript. Moreover, all figures and tables generated within the quarto document are not included in the Google Doc file, making it hard for collaborators to grasp the full content of the paper. Note that there is an ongoing discussion on implementing a collaborative platform for Quarto (https://github.com/quarto-dev/quarto-cli/discussions/405).

Moreover, there exist some tools within the R universe that I have not tried out yet but which seem to provide some benefits for reproducibility:

- *targets*: A pipeline tool for computationally demanding projects

- *rocker:* A tool to package an R program and all dependencies into a container. Compared to *renv*, it also takes care of dependencies that are not R packages.

3.6. Example for spreadsheet data: ODYM Data Processes (ODP) and the RECC model traceability steps

ODYM – The Open Dynamic Material Systems Model is an open source framework for material systems modeling programmed in Python. The description of systems, processes, stocks, flows, and parameters is object-based, which facilitates the development of modular software and testing routines for individual model blocks (Pauliuk and Heeren 2020). ODYM MFA models can handle any depth of flow and stock specification: products, components, sub-components, materials, alloys, waste, and chemical elements can be traced simultaneously. ODYM features a new data structure for material flow analysis, based on a newly developed data model for industrial ecology research (Pauliuk et al. 2019). All input and output data are stored in a standardized file format and can thus be exchanged across projects.

The databases and model calculations for ODYM-based projects can be complex, such as in the RECC (Resource Efficiency – Climate Change mitigation framework) model for circular economy scenarios (Pauliuk et al. 2021), which is why custom procedures for complete traceability of workflows and reproducibility of results were designed.

3.6.1. The general steps for complete traceability of workflows and reproducibility of results in the RECC model framework.

Figure 6 visualizes the traceability steps for the RECC model. While the workflow for data and result transparency goes from the original data sources to the final result documented in papers or reports, the tracing of results and data back to the original sources or model configuration goes the other way.

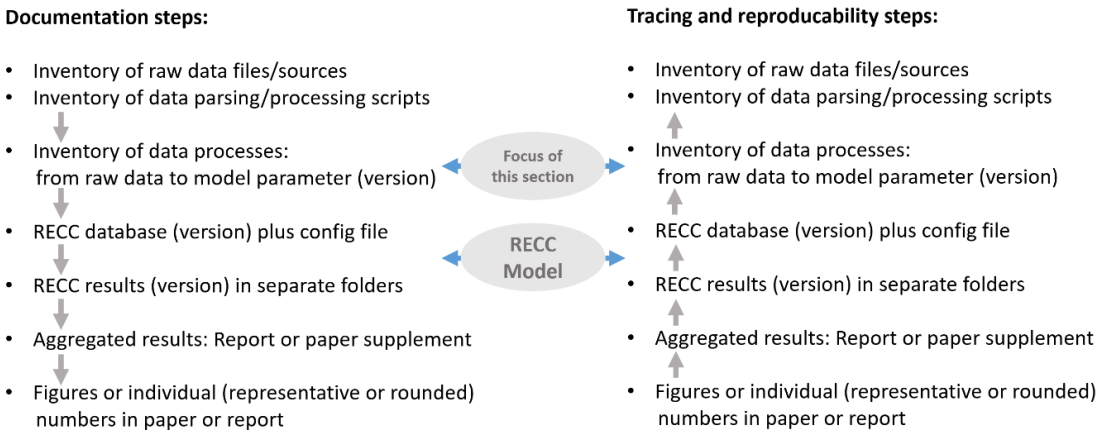

Fig. 6: Workflow with steps for complete traceability of numerical results in the RECC model.

The data transparency workflow was designed from the right side in Figure 6, by asking, at each research step: "Have I documented all the information to trace the results back to their original source and to reproduce my findings?" and documenting the information accordingly. Then, in the research practice, the workflow is the opposite, as shown on the left side of the figure.

In Box 5, we describe the implementation of the general steps for complete traceability of workflows and reproducibility of results in the RECC model for dynamic MFA.

Box 5: Implementation of the five general steps for complete traceability of workflows and reproducibility of results in the RECC model for dynamic material cycle scenario analysis

In the RECC model database, each model parameter is stored in a separate .xlsx file that follows a certain template structure, the elements of which are described below.

1. **Ensure the traceability of the original data used**

Each model parameter file has a reference sheet, where the links to the DOI or other identifiers of the used data are listed and described (like 'Table 3 in DOI xxx' or 'Supplementary Figure X for article DOI xxx'). Where suitable, the original data files are archived in a separate raw data archive, which is part of the project's hosting institution's research data management, and the relative path of the archived raw data file is documented on the reference sheet.

2. **Document how processed data was derived from original data**

Each model parameter file has a log sheet, where the ODYM data process (ODP, see documentation below) documents exactly how the different source data are converted (copied, reformatted, with or without additional assumptions or ancillary calculations) into the parameter values. The OPD documentation links to the version number of the parameter file.

3. **Keep track of how processed data are combined into datasets that form the model parameters**

Each model parameter file has a unique filename and a version number, which are listed in the RECC model's config file, so that each model configuration, which consists of 100+ model parameter files, is documented as a list of parameters including their exact versions.

4. **Document relevant calculations and model runs**

Each RECC model run creates a UUID and a separate result folder, where the current model configuration file (and with it the complete parameter and version list) is saved as a copy, the used model code version is documented (via Git commit ID), and a result summary is archived.

5. **Link final outputs to the calculations that produced them**

Result figures and tables are created by scripts and config files that are part of the RECC Git and Zenodo repositories. This way of structuring the result evaluation allows to trace the data flow from the result folders to the figures and tables. Each report or paper has a log table (part of the .xlsx workbook with all the plot and table data) that exactly documents/traces how the figures and tables were created and how each single quantitative statement in the paper was derived from the detailed quantitative results.

The details of the different steps follow from the specific setup of the RECC model structure. A central characteristic of the RECC model database is the use of data with highly varying volumes (from a single number to hundreds of thousand of parameter values in a single file) from a wide and very diverse set of data sources, ranging from specific tables in journal papers to ancillary model calculations with their own specific documentation.

Step 1 is thus straight forward: All these raw data sources need to be listed and intermediate steps (such as Excel workbooks supplied by colleagues or difficult to access supplementary material) be archived. See Figure 7 for a screenshot of a sample reference sheet of a RECC model parameter file.

literature_id	literature_key, dataset or sc	iedc_datase	iedc_dataset_vers	authors	title	year	journal_outl	city	DOI (for pap	URL or link to document, n	copyright	notes	archive location
1	Hotmaps building stock database		Hotmaps building st	Simon Pezzutto, Stefano Z	HOTMAPS D2.3 WP2	2018						for EU28	archived under: \RECC_D
2	Hotmaps and JRC_Nemry data extraction		EU_NonResBuildin	Pauliuk								for EU28	archived under: \RECC_D
3				CBECS Survey	Stock of non-residentia	2016				https://www.eia.gov/consumption/com		for USA	archived under: \RECC_D
4	USA nrb stock data workbook			PB, LP	USA nrb stock data workbook							for USA	archived under: \RECC_D
5				Lixuan Hong, Nan Zhou, D	Modeling China's Buildi	2014	Lawrence Berkeley National Laboratory			https://www.aceee.org/file	©2014 ACEE	for China	
6	Survey of Commercial and Institutional Energy Use (SCIEU) - Buildings 2014				Stock of non-residentia	2014				https://oee.nrcan.gc.ca/corporate/statis		for Canada	
7	Canada nrb stock data workbook, sheet SCIEU-2014			LP, SP	Canada nrb stock data 2021/22							for Canada	archived under: \RECC_D
8				Kumar et al.	Estimating India's comr	2019	Building Research & Inforn		https://doi.o	https://doi.org/10.1080/09613218.2018		for India	
9				SP	Sheet 'India' in this file	2022						for India	this file
10				GIZ		2019				https://www.giz.de/de/downloads/giz20		for Turkey/R5.2	MNF_Other
11				SP	Sheet 'Single_Number_							for Turkey/R5	this file
12				Chatellier-Lorentzen		2020	Sustainable Cities and Society			https://doi.org/10.1016/j.scs.2020.1021		for Mexico/R5.2	LAM_Other
13				SP	Sheet 'Single_Number_							for Mexico/R5	this file
14	RECC parameter 2_P_RECC_Population_SSP_32R, V2.2, ODP			UUID 76481f7c-db79-4d96-af8d-d48842f7c215								for R5.2	REF_Other
15				Okumura, K., Ikaga, T., Ka	Development of the dat	2012	AIJ J. Technol. Des. Vol. 1		DOI: 10.31	https://www.scopus.com/record/display		for Japan	
16				SP	Sheet 'Japan' in this file	2022						for Japan	this file

Fig. 7: Screenshot of the *ref* (reference) sheet of a RECC parameter file.

The crucial *step 2*, the logging routine for the different RECC model parameters, is documented below.

In *step 3*, the parameter list in the RECC model's config file is defined to select the right set of parameters and their exact versions. The config file lists the parameter file names so that there is a documented link between each model configuration and a specific model database.

In *step 4*, the model, when executed, documents its own run, by creating a UUID and a separate result folder for each single model run, where the valid model config file is saved (as a copy, and with it the complete parameter and version list), the used model code version is documented (via the Git commit ID), and the result summary is archived. A scenario definition file links all scenario runs that are defined and executed by the RECC model scripts with their corresponding result folders, so that a 1:1 match between each scenario definition (consisting of model version, database version, and model config settings) and each result folder is established.

In *step 5*, the result evaluation and documentation, model result figures and tables are created by scripts and scenario evaluation config files that are part of the RECC Git and Zenodo repositories. These RECC results evaluation scripts have built-in features that document (in RECC results evaluation log files) exactly which model results were analyzed by which scripts and what the exact results (figure and table file names and time stamps) are. Each report or paper has a log table (part of the .xlsx workbook with all the plot and table data) that exactly documents/traces how the figures and tables were created (from the RECC results evaluation log) and how each single quantitative

statement in the paper was derived from the detailed quantitative results. As the RECC model is being developed, more and more steps are automated to reduce manual and error-prone documentation work.

3.6.2. Linking original data to processed data with the ODYM Data Processes (ODP)

The ODYM Data Processes (ODP) represents a specific implementation of *step 2* in the general framework, the link of each single data point to its original data sources, and the documented data processing steps in between.

In the RECC model, the more than 120 individual parameter files consist of lists or tables with a multi-index to describe the meaning of the numbers. Fig. 8 below, for example, shows the data table for a building stock parameter, where the building stock (in million m²) is specified by year and region and broken down into age-cohorts and different building types. The data comes in table format but it has a hierarchical multi-index for both rows and columns. In a parameter that gives the stock for many different regions and types, a larger number of different data sources will be used to obtain the different parameter values. Fig. 8 shows a typical situation, where the individual numbers are marked with different colors, depending on what sources they come from.

For the different colors, not only the sources vary but also the way the data were processed. For example, it may be that one source has the data in the right resolution so they just need to be copied over. Another data source may only contain aggregate information so that proxy data have to be used in addition to arrive at the required level of resolution level. For still other figures, no matching data source may be available at all, so that proxy data have to be used or assumptions have to be made.

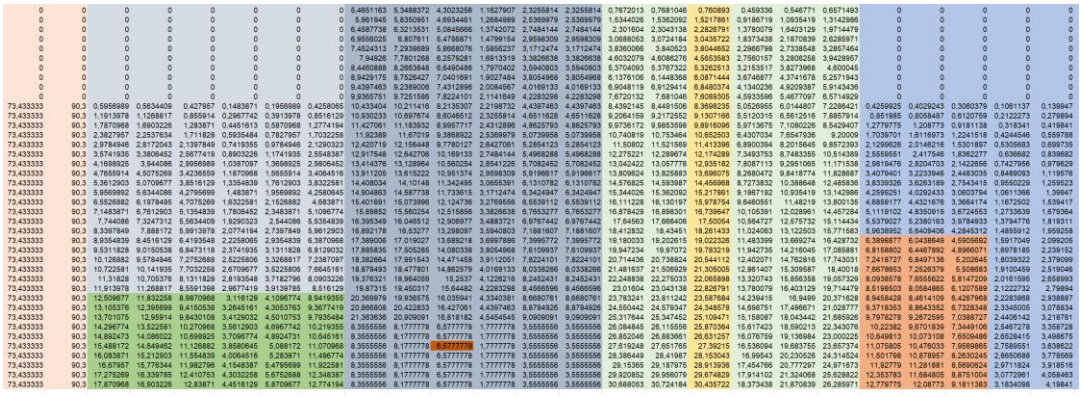

Fig. 8: Screenshot of the data sheet of a RECC parameter file, with the different colors indicating the different data sources and data processes they are linked to.

That means that the color code (or any other identifier linked to each individual data point) cannot just simply point to a data source but needs to point to a procedure, which in turn may involve a number of data sources, ancillary calculations, and

assumptions. Since the data format of the RECC model is that of the ODYM MFA software, the traceability documentation applies to all data in ODYM format, and we define the formalization of this traceability, the *ODYM data process (ODP)*, as follows:

Definition: *An ODYM data process (ODP) is a research procedure that links data from one or more data sources together with ancillary calculations, aggregation or disaggregation, and/or assumptions to output data that are contained and formatted in a parameter file in ODYM format.*

Below, the requirements and guidelines for the documentation of ODPs are given.

The following requirements were central for designing the ODP and implementing it in the RECC project:

- Need for a traceable and distributed (across different team members) workflow for different raw data that flow into RECC model parameters.
- The data documentation workflow needs to allow for different parameter versions to be branched and later merged from certain departing file.
- Each single number must be traceable.
- Flexible procedure from large datasets (100000+ data points from a single ODP) down to individual numbers.
- Model parameter files will contain a patch of data from different data processes, plus modifications and corrections for individual numbers.
- The link between each single number, the corresponding data source(s), and the data processing must be documented and traceable.
- The whole documentation process should be lean and not create much overhead.
- The ODPs need a machine-readable documentation, to facilitate automatic generation of reports, that is easy to read and edit for humans.

From these requirements, we derived the following specifications for documenting ODYM data processes (ODPs):

ODPs are documented in three parts with the following details:

(1) The *ref* sheet of an ODYM parameter file (see Fig. 7 for a sample) lists traceable links to all data sources (main and ancillary) as well as links to all additional calculation tools such as spreadsheets or scripts. These links can be GitHub links, paths to a local archive, or links to other sheets contain in the same .xlsx ODYM parameter file.

(2) The *log* sheet of an ODYM parameter file (see Fig. 10 for a sample) lists a definition and description of the different ODYM data process (ODP) used to compile the entire dataset (see details below). Each ODP is defined by a UUID and by a color.

(3) The *data* sheet of an ODYM parameter file (Fig. 8 for a sample) links each data point to the respective ODP using machine-readable color codes.

With these three parts, the parameter's data points are linked to the original sources (workflow direction, Fig. 9, top). In the retrieval direction (Fig. 9, bottom), the lookup direction is the exact opposite of the archiving direction.

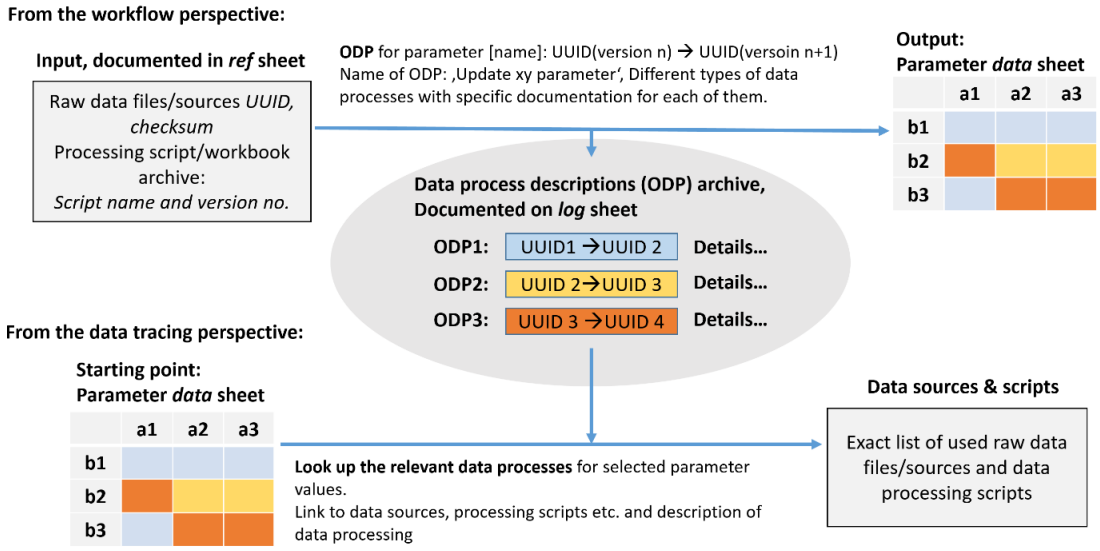

Fig. 9: ODYM data processes (ODP): General documentation scheme with three parts in the *ref*, *log*, and *data* sheets of the individual ODYM parameter files.

The central documentation place is the *log* sheet of the ODYM parameter file (see Fig. 10 for a sample). Here, the different ODPs are defined (name, type, UUID, color code), linked to metadata (date stamp, parameter version, name of researcher), and a detailed description is provided together with a link to all references used (data sources, scripts, etc.) as listed on the *ref* sheet.

ODP_Short_name	Date of ODP	Version number before ODP	Version number after ODP	old UUID	new UUID and reference color	Who	What - detailed description	ODP type	List items of ref sheet used
Create dataset and Transfer EU country data	21.11.2019		V1.0		a123e321-93de-453a-ea33-888f2f4df3e	sp	datafile created and filled with EU data from intermediate file RECC_Dataset_NonResBuildingArea_EU_9R extracted with script	0, 1	[1,2]
							EU in 9 regions: For Standard nonres. buildings: from Hotmaps project, ref. [1], converted and interpolated to all years by script ref. [2]		
Transfer USA data	06.04.2022	V1.0	V1.0	a123e321-9	eab5be57-1963-40ab-8bbb-6eaaad808a7	Peter Berrill, LP, SP	For the USA, detailed nonres building stock data are available for 2012 (10 age-cohort groups x 13 purposes, ref [3]). These data were aggregated to the 6 building groups used here and the age-cohort groups where split into the individual years. All data are multiplied by 0.093 to convert from sqft to m². We use the 2012 stock data for 2015 as proxy with same age structure but a slighty lower total stock (82000 instead of 87000 Msqft). To fill the 2013-2015 gap, we replicated age-cohort 2012 values for 2013-2015 to obtain a rough estimate. This process is documented in ref. [4].	3	[3,4]
Estimate China data	06.04.2022	V1.0	V1.0	eab5be57-19	34233199-85cf-4c15-8802-61a98b8cc68	sp	Fig. 2 in ref. [5] reports ~14750 million m² in 2015 non-res buildings, with the following breakdown: (units: million m²) (estimated from ruler)	2	[5]
							These data are allocated to standard efficiency buildings. Most of these buildings are assumed to be less than 30 years old, Hence, the totals are simply split evenly over the 1985-2015 age-cohorts.		
Transfer Canada data	06.04.2022	V1.0	V1.0	34233199-8	69dce5c5-dde1-4308-a660-99955b75d67	Peter Berrill, LP, SP	For Canada, detailed nonres building stock data are available for 2014 (8 age-cohort groups x 10 purposes, ref [6]). **Assumptions for 2014 table** 1. All years contribute equally in a time period 2. Office buildings (non-medical) aggregated with Medical office building in "Offices" 3. Commercial is comprised of food or beverage store and non-food retail store 4. Education is comprised of elementary and secondary schools only as universities are included in oher in the Canadian data 5. Health is comprised of Assisted daily/residential and hospital 6. Hotels and restaurants is comprised of Hotels, motels or lodges as there is no data on restaurants in the Canadian data 7. "Other" gathers Other from the Canadian data and Warehouse To fill the 2015 gap, we replicated age-cohort 2014 values for 2015 to obtain a rough estimate. This process is documented in ref. [7].	3	[6,7]
Estimate India data, use as proxy for R5.2ASIA	06.04.2022	V1.0	V1.0	69dce5c5-dc	42cfd59f-ca4b-444c-823e-9774ecb9223e	SP	Estimate data for India from 2017 data in journal paper, split into different age-cohorts, documented on sheet "India" in this file. Also, use the Scale to Other_ASIA with population ratio 963/1305	3	[8,9]
Estimate Turkey data and use as proxy for R5.1	10.04.2022	V1.0	V1.0	42cfd59f-ca	99fc4efa-047e-428d-8847-03b29d04041	SP	"Non-residential, commercial and public buildings cover a total of 0.5 billion m2 area (in 2015)" From ref. [10]. Estimate data for Turkey and use as proxy for R5.2MNF_Other and R5.2OECD_Other from 2019 data in GIZ report (ref. [10], split into diffe	2	[10,11]
Estimate Mexico data and use as proxy for R5.	10.04.2022	V1.0	V1.0	99fc4efa-04	81dd5056-2506-4086-a810-a63bfaa0e9a	SP	"with an estimated floor space of 272 million m2, accounting only for restaurants, offices, shops, supermarkets, theatres and hospitals." Fro Estimate data for Mexico and use as proxy for R5.2LAM_Other from data in journal paper by Chatellier-Lorentzen (ref. [12], split into different age-cohorts, documente	2	[12,13]
Use R32EU12-M as proxy for R2.5REF_Other	10.03.2022	V1.0	V1.0	81dd5056-2	7155b029-e8ce-4878-8a79-61178911cef	SP	Use R32EU12-M as proxy for R2.5REF_Other on a per Capita basis, use population from RECC parameter 2_P_RECC_Population_SSP2. Data for R32EU12-M, SSP2, IIASA GDP scenario, Million: 33,86583882	2	[14]

Fig. 10: Screenshot of the documentation of the ODP on the *log* sheet of a RECC parameter Excel template. The figure shows the documentation of the data processes, with UUID, type, color code, links to references, and documentation.

All together, this information must be complete and detailed enough and correct to allow the data users to link each single data point to the correct ODP via the color

code, the ODP to all references (data, scripts, workbooks, …) used, and provide enough verbal detail and ancillary calculations to make the data processing transparent.

Different types of ODPs were defined to make documentation easy and systematic.

To structure the documentation process that provides the verbal detail and ancillary calculations, different types of ODPs are used, depending on various situations for the compilation of model input data from a wide variety of sources with different methods. These situations include, but are not limited to: data transfer via script, manual transfer of a few individual numbers, data transfer with ancillary calculations in spreadsheets, assumptions, and more. See Table 1 for details of the different ODP types and specific documentation guidelines.

Table 1: Overview of the different ODYM data processes (ODP) types defined so far.

No.	ODP Type	Documentation details
0	**Create Parameter**	UUID and initial version number (1.0)
1	**Data-script-Parameter (for 'large' data)**	Script (list with path in *ref* sheet) parses raw data (list with DOI etc. in *ref* sheet), structures them (select, aggregate, …) and saves resulting data to parameter file.
1a	**From RECC scenario target table**	For the RECC model, a scenario target table with an interpolation script is used (Fishman et al. 2021). For parameters generated with this setup, always document the version number of the interpolation script and of the scenario target table.
2	**Small data**	A few individual numbers are extracted from the raw data (list with DOI etc. in *ref* sheet) and processed locally (documented directly in log sheet, with additional assumptions)
3	**Medium data**	Substantial reformatting of raw data is needed, documented on a separate workbook or worksheet (list with path in *ref* sheet).
4	**Assumption**	Mere assumption without additional references.
5	**Correction**	Correction of dataset, e.g., calculations or assignment of labels.
6	**Reformatting**	Reformatting of dataset, typically with a spreadsheet calculator. E.g., add or remove a new aspect, change the aspect order, etc., without additional data or aggregation/disaggregation. Used also when large data sets are manually reformatted (Excel…) after they were imported from other models.
7	**By definition**	Given by definition, for example, a correspondence table indicating that the provinces of a country are part of the country.

Large data are usually imported by a script, and any modifications and filtering of data is coded/documented directly in the script or the accompanying configuration file (if any). For this case, no new ODP type is defined, instead type 1 (data from script) shall be used.

3.7 Example for spreadsheet data and Monte Carlo Simulation with ODYM: MaTrace-multi and MaTrace-dissipation by Christoph Helbig, University of Bayreuth

An adaptation of the ODYM dataflow was implemented by Christoph Helbig at the University of Bayreuth for the two models MaTrace-multi and MaTrace-dissipation. (See Nakamura et al. (2014) for details on the MaTrace approach to modelling material cycles.) The goal for these models is to trace global metal flows from extraction to dissipation and calculate the longevity and circularity of the metals in the global economy. In MaTrace-multi, seven major metals are traced simultaneously in a one-region planetary model, allowing to identify issues of mixing and contamination. In MaTrace-dissipation, 61 metals are traced individually in a one-region planetary model, allowing us to compare the lifetimes of a large range of metals in the economy. The additional challenge during the implementation of this model was that the dynamic Material Flow Analysis was meant to include a Monte Carlo (MC) Simulation to estimate the uncertainty of key results based on the uncertainty of input parameters. The two models emerge from strong collaborations with other researchers, including Yasushi Kondo and Shinichiro Nakamura from Waseda University, Tokyo, Japan, and Alexandre Charpentier Poncelet, Université de Bordeaux, France.

In principle, the dataflow of MaTrace-multi and MaTrace-dissipation follows the ODYM data process (ODP) from Excel-based datasheets that provide the input parameters using the ODYM *data* sheet templates. Every single datapoint is commented with information on the data source and provided with a *stats array string*[16] that provides a condensed information on the underlying uncertainty distribution that will be assumed for the Monte Carlo Simulation.

The following uncertainty distributions are considered (with required parameters):

- Lognormal distribution (location, scale): ideal for physical quantities and monetary values.
- Weibull distribution (offset, scale, shape): ideal for lifetime distributions.
- Beta distribution (alpha, beta): ideal for efficiency, yield, and collection rates.
- Dirichlet distribution (multiple alpha values, one beta value): multivariate Beta distribution, ideal for allocation and split of material flows. (Note that stats arrays don't intrinsically support Dirichlet distributions, so this was adapted from the beta distributions in this model.)

The *python* code for MaTrace-multi and MaTrace-dissipation follows this structure:

[16] https://stats-arrays.readthedocs.io/en/latest/index.html

1. Load python libraries, create results path with timestamp, copy model and parameter files, read ODYM configuration, and define model classification.

2. Read parameters, set up MFA system, and initialize flow and stock variables.

3. Setup Monte Carlo Simulation to reduce computation time: Because drawing random numbers is computationally time-intensive, we want to draw random numbers in batches. (Example: it is computationally much faster to use *scipy.stats.norm.rvs(loc=mean, scale=std_dev, size=samplesize)* than to use *[scipy.stats.norm.rvs(loc=mean, scale=std_dev) for x in range(samplesize)]*). Therefore, all uncertain parameter arrays that have been created during reading the parameter files are subject to random sampling and stored into an array with one additional dimension "*sample*". Note that this creates arrays of higher dimensionality and significantly increases RAM usage of the ODYM code because usual sample sizes in Monte Carlo Simulation are 1000 or 10000 for statistical reasons.

4. Pre-calculate survival tables for lifetime distributions to reduce computation time: Just as it is computationally efficient to draw random samples in batches, it is also useful to evaluate lifetime distributions for all possible cohort-ages before the actual material flow calculations. This is already part of the ODP and has been taken over to MaTrace-multi and MaTrace-dissipation.

5. Calculate material flows and stocks without uncertainty using the mean values of parameters.

6. Save *pandas* dataframes for material flows and stocks without uncertainty as Excel result sheets in the previously created results path.

7. Create graphs from dataframes using *matplotlib*.

8. Calculate material flows and stocks with uncertainty using previously generated samples of parameters. This requires iterative calculations, one parameter set at a time. It is efficient only to save key results to dataframes that are later evaluated with percentiles to save memory usage for large dataframes.

9. Calculate percentiles for key results: For key results, calculate 2.5, 16, 50, 84, and 97.5 percentiles to get information on confidence intervals. Save these percentile dataframes to Excel result sheets. If useful, save mean results, too.

10. Create graphs from dataframes using percentiles using *matplotlib*.

By following this calculation procedure, a single result folder is created that contains all information from parameter files in ODYM *data* format, python code for the specific stock and flow model (here: *ODYM-MaTrace-multi.py* or *ODYM-MaTrace-*

dissipation.py), the ODYM packages *classes* and *functions*, the result dataframes in Excel sheets, and vector graphs created with *matplotlib*.

For publication, for both MaTrace-multi and MaTrace-dissipation, the code with input and output data and graphs has been fully uploaded to *Open Science Foundation* repositories at the time of acceptance of the corresponding journal articles in the version of the accepted manuscript, registered as an archived version, and a DOI has been assigned. Therefore, the full dataset and model is published *open access*.

Related peer-reviewed publications and OSF registrations:

Charpentier Poncelet, A., C. Helbig, P. Loubet, A. Beylot, S. Muller, J. Villeneuve, B. Laratte, A. Thorenz, A. Tuma, and G. Sonnemann. 2022. Losses and lifetimes of metals in the economy. Nature Sustainability 5(8): 717–726.

Helbig, C. and A. Poncelet. 2022. ODYM-MaTrace-dissipation. Open Science Framework, August 30. https://osf.io/cwu3d/ Accessed March 7, 2023.

Helbig, C., Y. Kondo, and S. Nakamura. 2022. ODYM-MaTrace-multi. Open Science Framework, August 30. https://osf.io/r54c6/. Accessed March 7, 2023.

Helbig, C., Y. Kondo, and S. Nakamura. 2022. Simultaneously tracing the fate of seven metals at a global level with MaTrace-multi. Journal of Industrial Ecology 26(3): 923–936.

3.8. The MISO2 model at the Institute for Social Ecology, Vienna: an example combining spreadsheets and python workflows, drawing on the ODYM ontology and model

The aim of the MISO2 model (dynamic integrated model of material inputs, stocks and outputs) is to cover economy-wide material cycles and stock dynamics across multiple materials and end-uses at the national to global level (Wiedenhofer et al., 2024). It was developed within the ERC Advanced Grant MAT_STOCKS.[17] The project focused on the role of material stock patterns for the transformation to a sustainable society, as most resource use and GHG emissions stem from building, maintaining and using stocks, which deliver fundamental services such as shelter, water and energy supply, mobility, and others. Stocks also create long-term lock-ins of resource use patterns (land, energy, etc.) and their efficiency of service provision is quite variable, resulting in substantial potentials but also limits for a sustainability transformation. MAT_STOCKS thereby aims to identify barriers and leverage points for future sustainability transformations and achieving the SDGs, and to elucidate the socio-ecological and political implications of transforming stocks and service provisioning (Haberl et al., 2019, 2017).

When developing MISO2, we aimed for a) long-term coverage of material cycles and stock dynamics around the world, b) coverage of multiple materials and end-uses, c) enabling linking these quantitative results to other analytical and qualitative research streams.

MISO2 features several innovations beyond the original MISO1 model (Wiedenhofer et al., 2019). First, material supply chain processes from raw material extraction, trade, processing, stock-building, use phase of material stocks, as well as waste collection, recycling and downcycling are explicitly represented, thereby clearly differentiating processes, which are linked by physical flows and which utilize stocks. Second, end-uses are differentiated, using a refined application of the 'end-use transfer waste input-output' method originally developed in (Streeck et al., 2023a, 2023b). Third, to understand major sources of uncertainty, one-at-a-time sensitivity and uncertainty testing is conducted, including Monte-Carlo Simulations for the end-use shares modelling, based on a systematic uncertainty assessment of the model input data developed previously (Plank et al., 2022a, 2022b). Fourth, MISO2 was implemented in the python programming environment and uses the ODYM data ontology and the dynamic stock model codebase (Pauliuk and Heeren, 2020). Finally, the model is empirically applied to quantify global, country-level stock-flow dynamics for 177 countries, 23 raw materials, 20 stock-building materials and 13 end-use product

[17] https://boku.ac.at/understanding-the-role-of-material-stock-patterns-for-the-transformation-to-a-sustainable-society-mat-stocks - This research was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (MAT_STOCKS, grant 741950).

groups, from 1900 to 2016 including a spin-up from 1820, yielding the MAT_STOCKS database currently in version 1.0

During the research effort, the system definition for MISO2 was continually refined, given data availability, the overall aims of the project, as well as ongoing developments regarding specific research questions and methodological possibilities, especially regarding end-use differentiations (Streeck et al., 2023a, 2023b). The final system definition covers 14 material supply chain processes and stock dynamics, fully consistent with the system boundaries established in economy-wide material flow accounting (ew-MFA); for the full system definition and further details see (Wiedenhofer et al., 2024). Below we give an overview of the workflows developed and how they relate to the general guidelines presented above. Overall, the development of the model and especially the model input data took ~5 years and combined contributions by multiple research assistants, PhD-students, post-docs, a dedicated data scientist, and a senior research scientist leading this effort. Each intermediate step of the process was published separately and involved substantial efforts in exploration, data scoping and experimentation.

3.8.1 Software architecture and workflows

The MISO2 model is implemented using Python packages combined into an integrated workflow (Figure 11). NumPy is used for numerical computations (Harris et al., 2020), SciPy for statistical computations (Gommers et al., 2024), Pandas for handling output (team, 2023), and Matplotlib (Team, 2023) / Seaborne (Waskom et al., 2022) for visualization. MISO2 uses the data ontology and components of the open dynamic stock-flow modelling package ODYM (Pauliuk and Heeren, 2020). MISO2 extends ODYM with components for automated input creation, input and output validation, uncertainty assessment and Monte Carlo simulations (Figure 11a). MISO2 was built in a collaborative GitHub repository incl. versioning and internal code review. A running version of the MISO2 code with some example data will be made open access accompanying the final publication of the scientific publication currently under review, which also contains definitions and equations for each process and sub-module of MISO2 (Wiedenhofer et al., 2024).

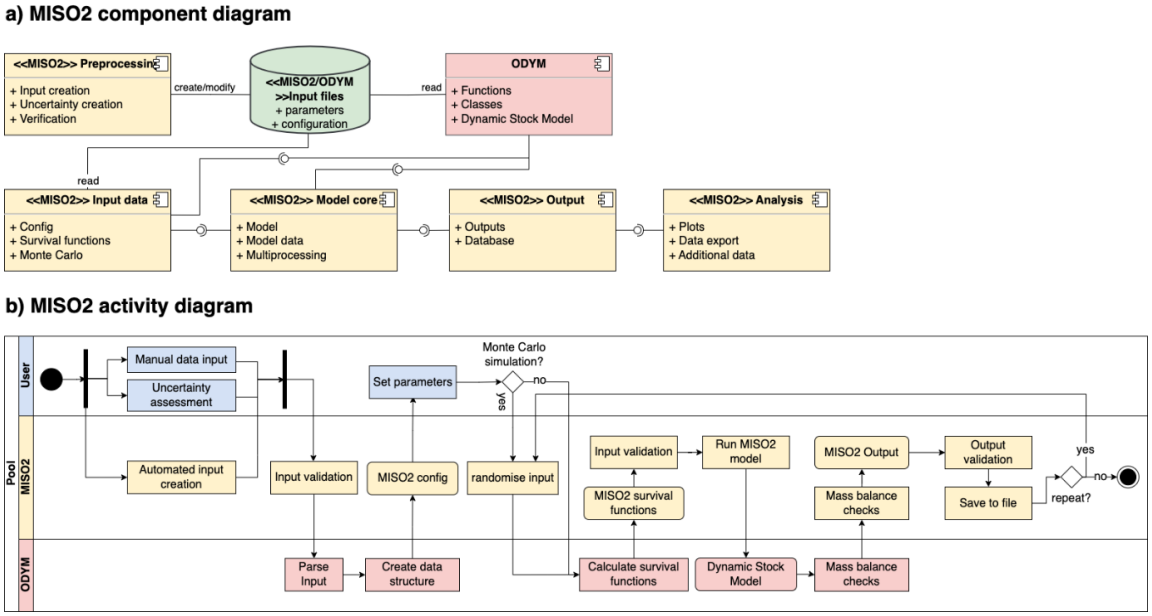

Figure 11: Unified Modelling Language (UML) component and activity diagram of model steps (Wiedenhofer et al., 2024).

To handle the high dimensionality of the input data, shorten processing times, and handle output aggregations, multi-processing via (McKerns, n.d.) and Dask is used (Dask Development Team, 2016). The MISO2 software package is covered by integration tests, with some of the core functionality further covered by unit tests. The input preprocessing of the end-use shares is handled via R scripts that make extensive use of routines and the Tidyverse library (Wickham et al., 2019). To ensure correctness of the results, data input and outputs are validated repeatedly through the entire workflow (Figure 11b). Validation encompasses both domain-specific methodologies such as mass balance checks and outlier identification, and generic approaches, such as the verification against negative or 'Not a Number' values.

The MISO2 workflow requires manual data pre-processing, preparation, and uncertainty assessment of the model input data, as well as parameter settings by the users (Figure 11a). These various data are then structured into the ODYM ontology and saved as a versioned model input dataset. The MISO2 software then automatically prepares the full model data inputs from partially sparse exogenous input data through interpolation and extrapolation (see below), validates its completeness, creates survival functions, computes material cycles and stock dynamics, validates all model data outputs, and saves into standardized data structures (Figure 11b).

3.8.2. Handling of exogenous model data inputs

The data pre-processing workflows and the uncertainty scoring were originally developed in (Plank et al., 2022a, 2022b), where we applied a ten-step procedure to develop a global-country level material flow database for primary materials extraction being processed into stock-building materials. For this purpose, historical yearbooks had to be digitized manually and combined with multiple international databases and

scientific as well as grey literature. The raw data are stored in excel workbooks and then further processed manually, using color coding to indicate inter/extrapolations, data sources, as well as assumptions taken. This formal 10-step procedure was consistently applied across ~20 materials, 177 countries and ~200 years, following conservative, but systematic and transparent rules explained in a separate methods paper (Plank et al., 2022b); results and detailed documentation are found in (Plank et al., 2022a).

Then, we developed additional exogenous model input parameters for each process and material as defined in the system definition, including losses and waste rates per process, recycling rates, downcycling rates, lifetime distributions, as well as end-use shares, following a formalized 6-step procedure aligned with the ten steps from (Plank et al., 2022b). These are: 1) data collection, 2) uncertainty scoring of each data point, 3) assigning collected parameters to our database structure prioritizing data with lowest uncertainties, 4) interpolation of data gaps & checks for plausibility, 5) extrapolation to non-available years, and 6) assigning uncertainty scores to inter/extrapolations and whenever assumptions were necessary.

3.8.3. Scoring and quantifying the uncertainty of model input parameters

We scored all model data inputs according to their reliability and fit to our system definition based on the evaluation framework proposed by (Laner et al., 2016, 2014) and operationalized in (Plank et al., 2022a, 2022b). Each datapoint was assessed along five independent data quality indicators and scored from 1-4. The criteria are reliability, completeness, temporal correlation, geographical correlation and further technological correlation; if those did not apply due to lacking (meta)data, expert judgment was used and scored from 1-4 based on the quality of the expert judgement.

Overall, this framework helps structure assessing the uncertainty of heterogenous data (sources) and assumptions in a consistent manner, given that many published data do not have “measured” uncertainty information. Still, some assumptions are required to translate scores into quantitative uncertainty ranges, meaning that resulting uncertainty ranges of model outputs do still depend on underlying choices and assumptions taken by the researchers.

Box 6: Five Step Summary for Ensuring Data Processing and Results Reproducibility in MISO2

1. Ensure the traceability of the original data used

Each model input parameter file has a separate reference sheet, containing raw data and pre-processed data in separate tabs, including color coded information regarding processing steps applied (see above). Complete traceability was not achieved, due to the use of historical data sources which had to be digitized manually. All original data files are archived in a separate raw data archive of the project's hosting institution's data storage. As a next step, we are currently developing more automated procedures to update the input data.

2. Document how processed data was derived from original data

Some data pre-processing occurred manually, using color-coded excel workbooks and generalized formal rules as well as partially heuristics (see 10-step procedure above). Other data pre-processing was done via Python, using the same generalized rules and additional integrity checks (see above). Each fully assembled model input dataset has a version ID and is separately stored to ensure reproducibility.

3. Keep track of how processed data are combined into datasets that form the model parameters

Each model parameter file has a unique filename and a version number, which are listed in the MISO2-ODYM config file. Each model configuration includes the model parameter files with their exact versions. In the automated workflow, input data in ODYM format is generated from a source file that references the original data sources and their assessments. The script automatically tracks the source for each data point when the input is created. Updates of the references are carried out automatically when data sources or interpolation rules are changed, allowing full traceability of input sources and transformations (Fig. 12).

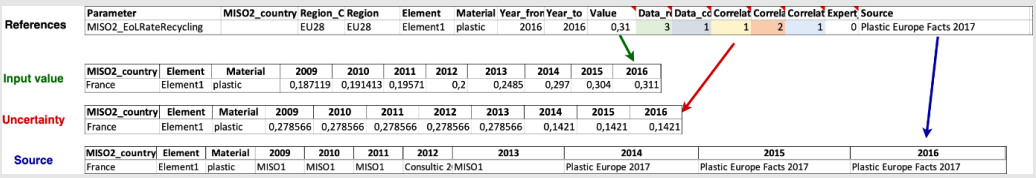

Figure 12: Setting input values, uncertainties and source information from literature references.

Box 6 ctd.:

4. Document relevant calculations and model runs

Each MISO2 model run creates a unique identifier and a separate result folder, where the current model configuration file (and with it the complete parameter and version list) is saved as a copy, the used model code version is documented (via Git commit ID), and a result summary is archived

5. Link final outputs to the calculations that produced them

Figures and tables are created by scripts and config files which are part of the internal extended MISO2 Github repository. Published results are provided as supplementary data files exactly as shown and are separately uploaded to Zenodo.

.

Core reference:

Wiedenhofer, D., Streeck, J., Wieland, H., Grammer, B., Baumgart, A., Plank, B., Helbig, C., Pauliuk, S., Haberl, H., Krausmann, F., 2024. From Extraction to End-uses and Waste Management: Modelling Economy-wide Material Cycles and Stock Dynamics Around the World. https://doi.org/10.2139/ssrn.4794611

4. Discussion and Outlook

The examples in this document do not represent a standard but a collection of good practice examples that shall inspire and trigger further development of reproducible research and traceable results.

The examples provided here have emerged bottom-up in different research groups. They have become part of the daily research practice, are accessible, and have become part of the training of early career researchers.

Below, we list and shortly describe other tools that may be worth exploring, discuss the further development of own tools and toolchains for our community, and describe the responsibilities of different agents in the research community.

4.1. Other tools for higher data transparency

During the compilation of this report, a number of other tools was reported to us but no concrete examples of their application in industrial ecology are documented. We list them here so that others can have a look.

- Simple *makefiles* can automatically handle steps 2-5 of the guideline (Box 1), see https://coderefinery.github.io/cmake/01-make-pipelines/ for an established pipeline
- There are also *full data pipeline frameworks*, including control and tracing of cloud computing and automatic scaling depending on workload, e.g., Prefect - https://docs.prefect.io/latest/ or Apache Spark. A curated list of such pipeline frameworks is available at https://github.com/pditommaso/awesome-pipeline.
- One of the most common data pipeline frameworks in research is *snakemake*, which was specifically developed to handle scientific data pipelines (https://snakemake.github.io/). Using such tools automatically handle steps 3-5 in Box 1. In particular, documenting data workflows in a structured way allows to version control the workflow steps, which includes tracing of model parameters and connecting them to data pipeline runs.
- For handling of distributed data across working environments, the *Dat* project may by interesting: https://docs.datproject.org/ "Dat is a protocol for sharing data between computers. Dat's strengths are that data is hosted and distributed by many computers on the network, […] that the original uploader can add or modify data while keeping a full history, and that it can handle large amounts of data. Datalad (https://www.datalad.org/) helps keep track of versions of data files and is basically an alternative to Dat.
- *Dolt* is a version management system for data in a database: "Dolt takes "Git for data" rather literally. Dolt implements the Git command line and associated operations on table rows instead of files. Data and schema are modified in the

working set using SQL. When you want to permanently store a version of the working set, you make a commit. Dolt produces cell-wise diffs and merges, making data debugging between versions tractable. Effectively, the result is Git versioning on a SQL database." See the following links for details:

https://www.dolthub.com/blog/2020-03-06-so-you-want-git-for-data/

https://github.com/dolthub/dolt

4.2. Tools for documenting SEM systems

SEM systems are often defined using a diagrammatic representation of the system processes and material types, with further written description of each process. Although this documentation can certainly be written by hand, there are some benefits to using tools to generate it, such as including automatic indexing of processes and material types defined in the system, and cross-references linking between them. The Sphinx documentation tool (https://www.sphinx-doc.org/) is commonly used for documenting computer software, generating HTML or PDF output from documents in Markdown or other simple formats, but it is very extensible to documenting entities in any domain. A Sphinx extension to an MFA system definition creates a description of "Processes" and "Objects" (i.e. types of material or good), allowing researchers to easily generate nice well-structured documentation of the structure of their SEM models, complete with indices and cross-references (e.g., https://github.com/ricklupton/sphinx_probs_rdf). This tool was originally conceived to produce machine-readable descriptions of the system elements for further processing (Germano et al. 2021) but can be used equally just to produce human-readable documentation for supplementary information to a study, e.g., in Malkovska et al. (2024).

4.3. Building our own community tools and toolchains

Our community needs a discussion to what extent standardized data structures and tools for basic MFA calculations are the key to effective, high quality, and traceable research in our field. Such a standardized toolchain would start from common classifications for materials, products, industries, regions, etc. (see here: https://github.com/IndEcol/SEM_classifications), include data formats and templates, standardized data processing and modelling routines, and standardized documenting of the research workflow. The software community for MFA that has evolved over the last years can contribute with the building blocks for such an effort: https://github.com/IndEcol/Dashboard?tab=readme-ov-file#material-flow-analysis, as well as the reproducibility/traceability examples presented here.

Education and capacity building

This guideline document and the collection of good practice examples are one step further towards higher traceability and reproducibility in our community. Further worked examples, customized and flexible software solutions, and trainings for reproducibility will be necessary and developed over the coming years. This material shall help spread the mindset and skills for traceability and reproducibility, and we expect it to be particularly useful for researchers and institutions that lack the resources to develop their own tools and workflows.

4.4. Responsibilities of individual researchers, scientific communities, funders, employers, and publishers

Traceability of results and reproducibility of claims are core scientific values, and all researchers need to strive towards their fulfilment to that science can fulfil its promise to society: to deliver accurate and reproducible knowledge.

There is also a major practical advantage that lies in traceable and reproducible research: It facilitate cumulative work and hand-over of projects in fluctuating teams. Often, the person trying to reproduce your work turns out to be yourself or your team. It is simply more professional to work this way!

Different actors in the scholarly world can contribute to traceable and reproducible research. The following list is taken from Pauliuk (2020):

All researchers need to be able to sense the transition from exploring new ideas or approaches ("trying out things") to proper research with documentation duties, and should switch to a professional mode that includes documentation for reproducibility and traceability.

For **early career researchers**, it becomes easier to connect to, use, and expand published material.

Editors, reviewers, and funders need to insist on method and data transparency, reproducibility, and proper data archiving.

Publishers should highlight and reward sound contributions to cumulative science and provide access to supplementary materials in any case and exclude it from copyright transfer (for pay-wall publishing).

Funders and employers also have a major responsibility here! These activities and standards need time and resources, which has major implications for project funding

which needs to allow for such work. Just think about the work that is required for updating time series and/or models, maintaining them, etc. – far too often, one has to cross-fund and/or sneakily insert those activities into fancy sounding work packages promising all kinds of novelty, instead of being able to position high quality reproducible workflows as a strength.

Employers/universities also need to understand these demands and give the appropriate resources, for example also having data scientist positions and proper IT support. To really get there, this needs institutional support, not only individual initiative.

Research associations, societies and scholarly communities should develop and adopt guidelines for reproducible and traceable research, transfer best practice from other disciplines, monitor research practice and available infrastructure, host research infrastructure where needed, facilitate exchange formats on cumulative research (workshops, special sessions at scientific conferences), and reward salient contributions to reproducible and traceable research.

Fully traceable research streams, built with professional tools, will be fun and effective! They will strengthen our research community and help with the application of our findings in society.

Acknowledgement

The authors gratefully acknowledge the exchange on reproducible workflows with numerous colleagues in the past, in particular:

Stefan Giljum (WU Vienna, Austria), who provided feedback on the guidelines.

Benedikt Grammer, who contributed to the graphics in section 3.8. He and Hanspeter Wieland provided some input to and feedback on section 3.8.

Bernhard Steubing (CML Leiden, The Netherlands), who provided examples of JIE papers that got awarded the data badge.

References

Allwood, J., Dunant, C., Lupton, R. & Serrenho, A. (2019). Steel Arising: Opportunities for the UK in a transforming global steel industry. Apollo - University of Cambridge Repository. https://doi.org/10.17863/CAM.40835

Barker, M., Chue Hong, N. P., Katz, D. S., Lamprecht, A.-L., Martinez-Ortiz, C., Psomopoulos, F., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., & Honeyman, T. (2022). Introducing the FAIR Principles for research software. Scientific Data, 9(1), 622. https://doi.org/10.1038/s41597-022-01710-x

Belmin, C., Hoffmann, R., Elkasabi, M. & Pichler, P.-P. LivWell: a sub-national Dataset on the Living Conditions of Women and their Well-being for 52 Countries. Sci Data 9, 719 (2022).

Belmin, C., Hoffmann, R., Pichler, P.-P. & Weisz, H. Fertility transition powered by women's access to electricity and modern cooking fuels. Nat Sustain 5, 245–253 (2022).

Boulay, A.-M., P. Lesage, B. Amor, and S. Pfister. 2021. Quantifying uncertainty for AWARE characterization factors. Journal of Industrial Ecology 25(6): 1588–1601. https://doi.org/10.1111/jiec.13173.

Charpentier Poncelet, A., C. Helbig, P. Loubet, A. Beylot, S. Muller, J. Villeneuve, B. Laratte, A. Thorenz, A. Tuma, and G. Sonnemann. 2022. Losses and lifetimes of metals in the economy. Nature Sustainability 5(8): 717–726.

Chen, W.-Q., Eckelman, M. J., Sprecher, B., Chen, W., & Wang, P. (2024). Interdependence in rare earth element supply between China and the United States helps stabilize global supply chains. One Earth, 7(2), 242–252. https://doi.org/10.1016/j.oneear.2024.01.011

Fishman, T., S. Pauliuk, N. Heeren, P. Berrill, Q. Tu, P. Wolfram, and E.G. Hertwich. 2021. A comprehensive set of global scenarios of housing, mobility, and material efficiency for material cycles and energy systems modelling. Journal of Industrial Ecology 25(2): 305–320. http://osf.io/preprints/socarxiv/tqsc3.

Germano, S., Saunders, C., Horrocks, I., & Lupton, R. (2021). Use of Semantic Technologies to Inform Progress Toward Zero-Carbon Economy. In A. Hotho, E. Blomqvist, S. Dietze, A. Fokoue, Y. Ding, P. Barnaghi, A. Haller, M. Dragoni, & H. Alani (Hrsg.), The Semantic Web – ISWC 2021 (S. 665–681). Springer International Publishing. https://doi.org/10.1007/978-3-030-88361-4_39

Haas, W., Virág, D., Wiedenhofer, D., & von Blottnitz, H. (2023). How circular is an extractive economy? South Africa's export orientation results in low circularity and insufficient societal stocks for service-provisioning. Resources, Conservation and Recycling, 199, 107290. https://doi.org/10.1016/j.resconrec.2023.107290

Haas, W., Krausmann, F., Wiedenhofer, D., Lauk, C., Mayer, A., (2020). Spaceship earth's odyssey to a circular economy - a century long perspective. Resour. Conserv. Recycl. 163, 105076. https://doi.org/10.1016/j.resconrec.2020.105076

Haberl, H., Wiedenhofer, D., Erb, K.-H., Görg, C., Krausmann, F., (2017). The Material Stock–Flow–Service Nexus: A New Approach for Tackling the Decoupling Conundrum. Sustainability 9, 1049. https://doi.org/10.3390/su9071049

Haberl, H., Wiedenhofer, D., Pauliuk, S., Krausmann, F., Müller, D.B., Fischer-Kowalski, M., (2019). Contributions of sociometabolic research to sustainability science. Nat. Sustain. 2, 173–184. https://doi.org/10.1038/s41893-019-0225-2

Harpprecht, C., L. van Oers, S.A. Northey, Y. Yang, and B. Steubing. 2021. Environmental impacts of key metals' supply and low-carbon technologies are likely to decrease in the future. Journal of Industrial Ecology 25(6): 1543–1559. https://doi.org/10.1111/jiec.13181.

Helbig, C. and A. Poncelet. 2022. ODYM-MaTrace-dissipation. Open Science Framework, August 30. https://osf.io/cwu3d/ Accessed March 7, 2023.

Helbig, C., Y. Kondo, and S. Nakamura. 2022. ODYM-MaTrace-multi. Open Science Framework, August 30. https://osf.io/r54c6/. Accessed March 7, 2023.

Helbig, C., Y. Kondo, and S. Nakamura. 2022. Simultaneously tracing the fate of seven metals at a global level with MaTrace-multi. Journal of Industrial Ecology 26(3): 923–936.

Hertwich, E.G., N. Heeren, B. Kuczenski, G. Majeau-Bettez, R.J. Myers, S. Pauliuk, K. Stadler, and R. Lifset. 2018. Nullius in Verba. Advancing Data Transparency in Industrial Ecology. Journal of Industrial Ecology 22(1): 6–17.

Jaccard, I. S., Pichler, P.-P., Többen, J. & Weisz, H. The energy and carbon inequality corridor for a 1.5 °C compatible and just Europe. Environ. Res. Lett. 16, 064082 (2021).

Kuczenski, B., C. Vargas Poulsen, E.L. Gilman, M. Musyl, B. Winkler, and R. Geyer. 2022. A model for the intensity of fishing gear. Journal of Industrial Ecology 26(6): 1847–1857. https://doi.org/10.1111/jiec.13156.

Laner, D., Feketitsch, J., Rechberger, H., Fellner, J., (2016). A Novel Approach to Characterize Data Uncertainty in Material Flow Analysis and its Application to Plastics Flows in Austria. J. Ind. Ecol. 20, 1050–1063. https://doi.org/10.1111/jiec.12326

Laner, D., Rechberger, H., Astrup, T., (2014). Systematic Evaluation of Uncertainty in Material Flow Analysis. J. Ind. Ecol. 18, 859–870. https://doi.org/10.1111/jiec.12143

Lupton, R & Serrenho, A. (2019a). ricklupton/uk-steel-trade: v2.0.0 (v2.0.0). Zenodo. https://doi.org/10.5281/zenodo.2591364

Lupton, R & Serrenho, A. (2019b). ricklupton/uk-steel-model: Initial release (v1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.2592184

Lupton, R & Serrenho, A. (2019c). ricklupton/steel-arising-report: Initial version (v1.0.0). Zenodo. https://doi.org/10.5281/zenodo.2592582

Malkowska, D., Lupton, R., Wellock, G., Boyle, S., Meng, F., Cullen, L., Cullen, J. (2024). Consistency and completeness of data on the global petrochemicals sector's emissions. Under review.

Mayer, A., Haas, W., Wiedenhofer, D., Krausmann, F., Nuss, P., & Blengini, G. A. (2019). Measuring Progress towards a Circular Economy: A Monitoring Framework for Economy-wide Material Loop Closing in the EU28. Journal of Industrial Ecology, 23(1), 62–76. https://doi.org/10.1111/jiec.12809

Meide, M. van der, C. Harpprecht, S. Northey, Y. Yang, and B. Steubing. 2022. Effects of the energy transition on environmental impacts of cobalt supply: A prospective life cycle assessment study on future supply of cobalt. Journal of Industrial Ecology 26(5): 1631–1645. https://doi.org/10.1111/jiec.13258.

Nakamura, S., Kondo, Y., Kagawa, S., Matsubae, K., Nakajima, K., & Nagasaka, T. (2014). MaTrace: Tracing the Fate of Materials over Time and Across Products in Open-Loop Recycling. Environmental Science & Technology, 48(13), 7207–7214. https://doi.org/10.1021/es500820h

Pauliuk, S. 2020. Making sustainability science a cumulative effort. Nature Sustainability 3: 2–4. http://www.nature.com/articles/s41893-019-0443-7%0Ahttps://www.nature.com/articles/s41893-019-0443-7.

Pauliuk, S., T. Fishman, N. Heeren, P. Berrill, Q. Tu, P. Wolfram, and E.G. Hertwich. 2021. Linking Service Provision to Material Cycles – A New Framework for Studying the Resource Efficiency-Climate Change Nexus (RECC). Journal of Industrial Ecology 25(2): 260–273.

Pauliuk, S. and N. Heeren. 2020. ODYM - An Open Software Framework for Studying Dynamic Material Systems - Principles, Implementation, and Data Structures. Journal of Industrial Ecology 24(3): 446–458. https://doi.org/10.1111/jiec.12952

Pauliuk, S., N. Heeren, M.M. Hasan, and D.B. Müller. 2019. A general data model for socioeconomic metabolism and its implementation in an industrial ecology data commons prototype. Journal of Industrial Ecology 23(5): 1016–1027.

Pauliuk, S., G. Majeau-Bettez, C.L. Mutel, B. Steubing, and K. Stadler. 2015. Lifting Industrial Ecology Modeling to a New Level of Quality and Transparency. A Call for More Transparent Publications and a Collaborative Open Source Software Framework. Journal of Industrial Ecology. 19(6): 937–949.

Pichler, P.-P., Jaccard, I. S., Weisz, U. & Weisz, H. International comparison of health care carbon footprints. Environ. Res. Lett. 14, 064004 (2019).

Plank, B., Streeck, J., Virág, D., Krausmann, F., Haberl, H., Wiedenhofer, D., (2022a). From resource extraction to manufacturing and construction: flows of stock-building materials in 177 countries from 1900 to 2016. Resour. Conserv. Recycl. 179, 106122. https://doi.org/10.1016/j.resconrec.2021.106122

Plank, B., Streeck, J., Virág, D., Krausmann, F., Haberl, H., Wiedenhofer, D., (2022b). Compilation of an economy-wide material flow database for 14 stock-building materials in 177 countries from 1900 to 2016. MethodsX 9, 101654. https://doi.org/10.1016/j.mex.2022.101654

SEM Board. 2021. Guidelines for Data Modeling and Data Integration for Material Flow Analysis and Socio-Metabolic Research. Freiburg, Germany. https://doi.org/10.6094/UNIFR/217970.

Steubing, B., A. de Koning, S. Merciai, and A. Tukker. 2022. How do carbon footprints from LCA and EEIOA databases compare? A comparison of ecoinvent and EXIOBASE. Journal of Industrial Ecology 26(4): 1406–1422. https://doi.org/10.1111/jiec.13271.

Streeck, J., Pauliuk, S., Wieland, H., Wiedenhofer, D., (2023a). A review of methods to trace material flows into final products in dynamic material flow analysis: From industry shipments in physical units to monetary input–output tables, Part 1. J. Ind. Ecol. 27, 436–456. https://doi.org/10.1111/jiec.13380

Streeck, J., Wieland, H., Pauliuk, S., Plank, B., Nakajima, K., Wiedenhofer, D., (2023b). A review of methods to trace material flows into final products in dynamic material flow analysis: Comparative application of six methods to the United States and EXIOBASE3 regions, Part 2. J. Ind. Ecol. n/a. https://doi.org/10.1111/jiec.13379

Vilaysouk, X., S. Saypadith, and S. Hashimoto. 2020. Preprint: Semi-supervised machine learning classification framework for material intensity parameters of residential buildings. July 30.

Wiedenhofer, D., Fishman, T., Lauk, C., Haas, W., Krausmann, F., (2019). Integrating Material Stock Dynamics Into Economy-Wide Material Flow Accounting: Concepts, Modelling, and Global Application for 1900–2050. Ecol. Econ. 156, 121–133. https://doi.org/10.1016/j.ecolecon.2018.09.010

Wiedenhofer, D., Fishman, T., Plank, B., Miatto, A., Lauk, C., Haas, W., Haberl, H., Krausmann, F., (2021). Prospects for a saturation of humanity's resource use? An analysis of material stocks and flows in nine world regions from 1900 to 2035. Glob. Environ. Change 71, 102410. https://doi.org/10.1016/j.gloenvcha.2021.102410

Wiedenhofer, D., Streeck, J., Wieland, H., Grammer, B., Baumgart, A., Plank, B., Helbig, C., Pauliuk, S., Haberl, H., Krausmann, F., (2024). From Extraction to End-uses and Waste Management: Modelling Economy-wide Material Cycles and Stock Dynamics Around the World. https://doi.org/10.2139/ssrn.4794611

Wolfram, P., Q. Tu, N. Heeren, S. Pauliuk, and E.G. Hertwich. 2021. Material efficiency and climate change mitigation of passenger vehicles. Journal of Industrial Ecology 25(2): 494–510.