ENBIOS

USER MANUAL

Version 1.2.11

October 2022

Nick Martin, Miquel Sierra, Alexander de Tomás, Rafael Nebot Medina and Cristina Madrid

Contact: Cristina Madrid (cristina.madrid@uab.cat)

Contents

1 Introduction

ENBIOS is a software system for the assessment of sustainability indicators of energy system configurations. Its first version was developed within the EU Horizon 2020 project SENTINEL, but it has been further developed within the CHIST-ERA project SEEDS, the Spanish Retos project LIVEN and the Horizon Europe project JUSTWIND4ALL. ENBIOS has expanded the Nexus Information System (NIS) developed as part of the Horizon 2020 project MAGIC-Nexus.

The purpose of ENBIOS is to compute a set of environmental impact and sustainability indicators by building a module that relates the various sources of energy within a system to the pressures caused by the technologies that provide them, integrating information and knowledge from three methodologies: life cycle assessment (LCA), the Multi-Scale Integrated Analysis of Societal and Ecosystem Metabolism (MuSIASEM) method and the dynamic simulation-optimisation of energy networks.

Once the required structural and data inputs are integrated into an ENBIOS simulation, the software allows various sets of specific data outputs (i.e., quantitative characteristics) to be derived for each energy source within a system. The structure of the system can then be used within a MuSIASEM framework (known as a “dendrogram”) to elaborate indicators that can be manipulated to provide deeper understandings of the inner functions of the system at various levels. This, in turn, allows the user to assess and compare the consequences that relate to the various elements within an energy system in a way that allows a better understanding and method of comparison for a range of future climate policy scenarios. The software has been employed within the SENTINEL project to analyse different future energy scenarios, particularly those that relate to the transition towards higher levels of renewable energy use.

While the core idea behind ENBIOS is relatively simple, implementing a model simulation requires a significant amount of information to be compiled and arranged before the software can be executed. Nevertheless, once the framework of the system and the basic system information has been defined, simulations for different scenarios can be undertaken by making relatively minor changes to system configurations and input files.

The following sections provide a concise description of the requirements for operating the software. Firstly, a brief description of the method for installing the software to a local device is provided. The required data inputs are then divided into three general categories and discussed in turn. Finally, the method for executing individual simulations, and the results that are produced, are outlined.

A simplified overview of the inputs, outputs and commands used by ENBIOS is given in Figure 1.

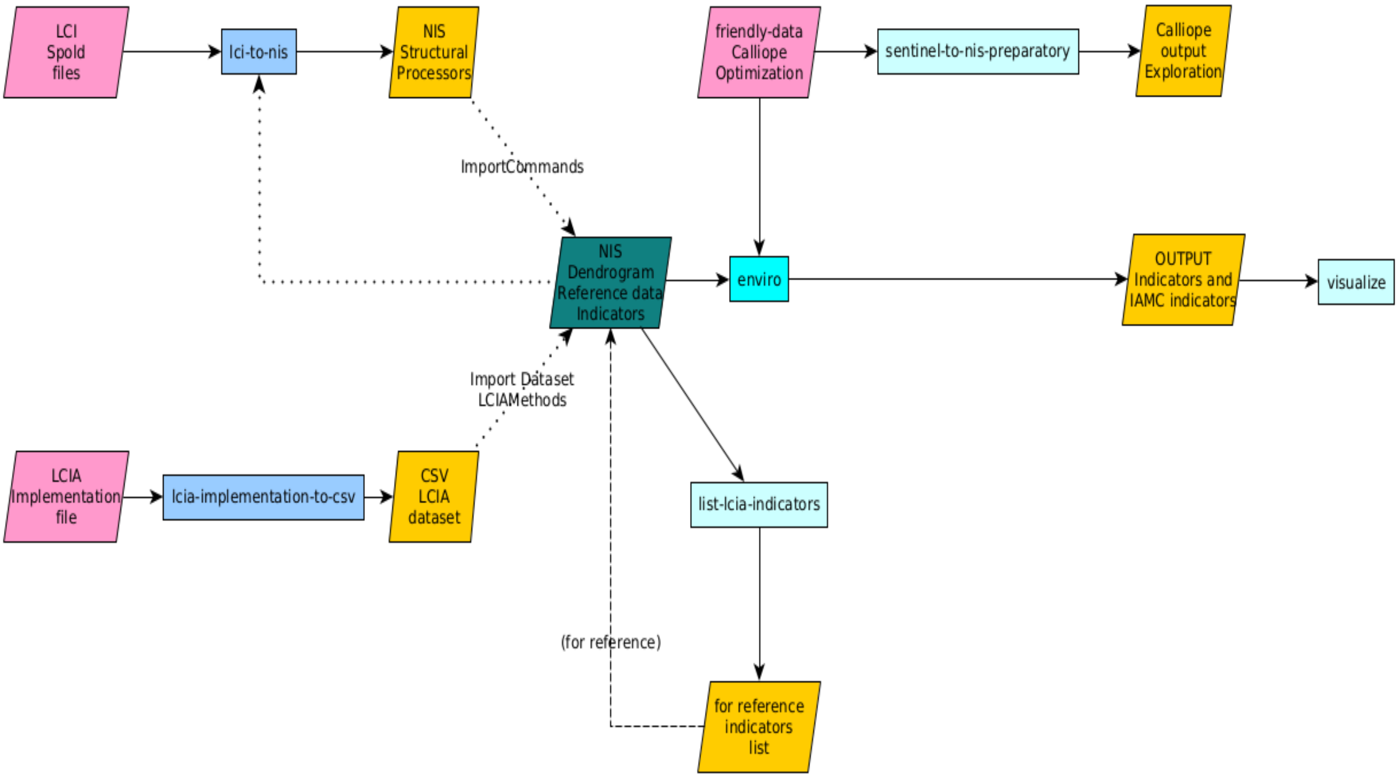

Figure 1 Inputs, outputs and commands used within ENBIOS

2 Installation of software

ENBIOS is a Python-based package. To install it, complete the following steps:

- Install a Python (CPython) >=3.7 runtime.

  Anaconda3 is recommended (https://docs.anaconda.com/anaconda/install).

- Open an Anaconda terminal:

  In Anaconda, select “Environments” > “base (root)”, then click the arrow symbol.

- Create a Python virtual environment:

  ```
  conda create -y -n enbios python==3.8 pycurl cython
  ```

- Activate the ENBIOS environment:

  ```
  conda activate enbios
  ```

- Install ENBIOS:

  ```
  pip install nexinfosys enbios
  ```

- To upgrade ENBIOS and Nexinfosys, when required:

  ```
  pip install --upgrade nexinfosys
  pip install --upgrade enbios
  ```

Once installed, an ENBIOS command should be available from the command prompt (CMD in Windows, Shell in Linux, Terminal in macOS). Simply executing this command without parameters (i.e., entering `enbios`) should produce a summary of the different actions. Here, the main action and final simulation function is known as ENVIRO; other actions within ENBIOS have the purpose of helping in the preparation of inputs, as discussed in the following section.

3 Preparation of inputs

The structural and data components required to successfully set-up and complete a simulation run in ENBIOS are divided into three general categories:

- Life cycle assessment data and methods
- General system setup
- Indicator definitions

These will each be discussed in detail in the sections that follow.

The software is designed to enable cooperative work on case studies by aligning with existing tools that provide collaborative functionality. When used as a Python package (not discussed in this document), using shared Python or R (rpy2 package) scripts or Jupyter notebooks is possible. However, perhaps a simpler way to collaboratively apply the software is via the use of shared Google Drive locations when operating ENBIOS actions from a command line. In particular, it is recommended that the “base” workbook file–the “mothership” document that defines the energy system structure and general model characteristics–is placed in a shared folder on Google Drive to allow different users to access common system definitions. Nevertheless, if collaboration is not required, local .xlsx files can be used as a “base” workbook. Most other input and output files should be placed on local computers.

3.1 Life cycle assessment data and methods

The key data inputs to most ENBIOS-based investigations will be provided by life cycle assessment (LCA) information. Two fundamental types of LCA inputs are required in order to prepare a simulation of ENBIOS. Firstly, the life cycle inventory (LCI) data–that provides the material and other inputs and outputs required to undertake individual energy-related processes within a system–must be converted from the native Ecospold (.spold) format, available from the Ecoinvent database (Ecoinvent, 2021; Wernet et al., 2016), into an ENBIOS-compatible file. Secondly, a list of the life cycle impact assessment (LCIA) methods–that define the ways that all of the individual listings within LCI datasets are transformed into specific indicator values–must also be converted into an ENBIOS-compatible file.

3.1.1 LCI inventory data

Data from the .spold files that define the inventories at each of the energy-related processes within the system can be converted into a NIS-compatible summary file using the `lci_to_nis` action. This requires the following parameters to be specified:

- `spold_files_folder` – Local path of the folder where all .spold LCI files are located
- `nis_base_url` – Location of "base" workbook file that defines the mapping information that links structural processors with .spold files. Note that if the `Accounted` property in the file is set as "No", the .spold filename specified for the specified `ParentProcessor` will be taken. It is assumed that this will be the URL of a shared online file. However, a local file can also be used, if prefixed by **Error! Hyperlink reference not valid.**
- `nis_structurals_output` – Local path where the output NIS file, in .xlsx format, will be generated

Usage:

```
enbios lci_to_nis nis_base_url spold_files_folder
nis_structurals_output
```

Example:

```
enbios lci_to_nis https://docs.google.com/spreadsheets/d/1Kkrbd-
GkaLtrJwH2xTFPoe217QqsZaiSfuFss-iUnsg/edit#gid=387453521 "C:\
ENBIOS\--- Selected Ecospold Files\v3.8 APOS LCI" "C:\ENBIOS\
lci_to_nis\lci_to_nis_output.xlsx"
```

When the process is completed, the resulting.xlsx file should be moved to the required shared folder (in case of collaborative work). This link should then be copied into the "`ImportCommands LCI Structurals`" sheet in the "base" workbook file (see Section 3.4).

3.1.2 LCIA methods

Within the LCA field, inventory data provided within LCI listings for individual processes can be converted into final indicators by using "methods" that apply various characterization factors to each inventory item. To integrate the chosen LCIA methods into ENBIOS calculations, an action is required to convert the raw LCIA method information–available alongside Ecoinvent database sources (Ecoinvent, 2021), typically as an Excel file named "LCIA Implementation__.xlsx", or similar–into a NIS-compatible file. This action is known as `lcia_implementation_to_csv` and requires the following two parameters to be specified:

- `method_file` – The path of the input .xlsx file containing LCIA method information
- `lcia_file` – The path of the output .csv file
- `method_like` – To select methods containing a given string
- `include_obsolete` – If "True", include methods marked as "superseded" or "obsolete"

Other "flags" can also be specified as part of a lcia_implementation_to_csv command:

- method_is – to filter specific LCIA methods.

Usage:

```
enbios lcia_implementation_to_csv method_file lcia_file
```

Examples:

- Without any flags:

```
enbios lcia_implementation_to_csv "C:\ENBIOS\
lcia_implementation_to_csv\LCIA Implementation v3.8.xlsx" "C:\
ENBIOS\lcia_implementation_to_csv\lcia_method_out.csv"
```

- To filter the method "ReCiPe Midpoint (H) V1.13":

```
enbios lcia_implementation_to_csv --method-is "ReCiPe Midpoint (H)
V1.13" "C:\ENBIOS\lcia_implementation_to_csv\LCIA Implementation
v3.8.xlsx"
"C:\ENBIOS\lcia_implementation_to_csv\lcia_method_out.csv"
```

- To filter two methods at the same time ("ReCiPe Midpoint (H) V1.13" and "cumulative exergy demand"):

```
enbios lcia_implementation_to_csv --method-is "['ReCiPe Midpoint (H)
V1.13', 'cumulative exergy demand']"
 "C:\ENBIOS\lcia_implementation_to_csv\LCIA Implementation
v3.8.xlsx"
"C:\ENBIOS\lcia_implementation_to_csv\lcia_method_out.csv"
```

The resulting .csv file will contain the following columns:

- `LCIAMethod` – Identifier of the LCIA method
- `LCIAIndicator` – Identifier of the LCIA indicator (e.g., "GWP" for Global Warming Potential)
- `LCIAIndicatorUnit` – The unit expected for the "Indicator" value
- `Interface` – The name of the exchange whose value will be picked from each considered activity (processor) for the weighting
- `InterfaceUnit` – The unit expected for the "Interface" value
- `LCIAHorizon` – When the ReCiPe method is chosen, the "cultural perspective" employed in the ReCiPe method according to Huijbregts et al. (2017):
  - "I" – Individualist
  - "H" – Heirarchist
  - "E" – Egalitarian
- `LCIACoefficient` – The weighting factor
- `Compartment` – Specifies to which environment compartment the weighting applies (e.g., water, air, soil, natural resource
- `Subcompartment` – Specifies the subcompartment of the above (e.g., water > surface water)

When the process is completed, the resulting .csv file should be moved to the required shared folder (in case of collaborative work). This link should then be copied into the "DatasetDef" sheet in the "base" workbook file (see Section 3.4).

Note that the above action only needs to be done when you wish to upgrade to a new version of the methods defined by the Ecoinvent database. Otherwise, you can continue to use the same version of `lcia_file` each time you use ENBIOS without running `lcia_implementation_to_csv` again.

3.2 General system setup

Again, when creating or altering ENBIOS simulations, the "base" workbook file–an Excel (.xlsx) or Google Sheets format file that contains several sheets that define the key aspects of the system–acts as the central "mothership" for the setup and operation of module simulations. An example of such a file can be found here. The definitions within this file largely follow the approach and nomenclature defined by the previous NIS software package. The various pieces of information contained within the sheets of this file are detailed in the five sections that follow.

It is noted that much of the information within the "base" workbook file will not be discussed in this document and, indeed, will not change from case to case once they have been initially specified. Nevertheless, full documentation of the approach adopted within the NIS software package–that provided the general basis of the development of ENBIOS–can be used to provide an overview of the file and how it functions (see Magic Nexus, n.d.).

Lastly, it is noted that entering a "#" symbol in the first column of a row causes ENBIOS to ignore that row (i.e., it is only for remarks or comments and is not used).

3.2.1 Overall energy system definition

The most fundamental setup information used within an ENBIOS simulation is the definition of the energy system schema (i.e., the hierarchical categorisation or framework of the system in question). Indeed, aggregations and relationships between contributing processes are ultimately based on the way in which the elements of the system are specified within this hierarchical definition.

The system is defined using the approach defined by MuSIASEM (Giampietro et al., 2009) whereby a system is defined by a "dendrogram" made up of a collection of elemental nodes known as "processors". Although the term is defined in more detail elsewhere (e.g., see di Felice et al., 2019; González-López and Giampietro, 2018), a processor essentially represents a single process that can be further defined as being either "functional" or "structural" in nature. Structural processors represent the most disaggregated level of the system. Here, a given technology performs a set of biophysical transformations to produce a profile of inputs and outputs for the given structural element. Meanwhile, functional processors are locations where data from structural (or higher-level functional processors) are aggregated and where further data for the metabolism of inputs to produce outputs can be obtained. To provide an energy-related example, one could define a series of structural processors for electricity produced from wind, solar, hydro and so on; all of these could then be aggregated in a functional processor at the previous level for electricity from renewable sources. Another structural processor at the previous level from this processor could aggregate all of the electricity produced from renewable and non-renewable sources. This approach is highly flexible and can be used to characterise almost any socio-metabolic system that can be defined into a hierarchical framework.

The energy system schema is defined within the "`BareProcessors top levels`" sheet of the "base" workbook file. The columns that define this are as follows:

- `Processor` – The name of the processor
- `ParentProcessor` – The processor, located one level above, where the data from this processor is aggregated
- `SubsystemType` – Type of subsystem that the processor is within, typically "Local"
- `FunctionalOrStructural` – Whether the processor is functional or structural
- `Accounted` – Turns accounting on ("Yes") or off ("No") for this processor, typically "Yes". If set as "No", the .spold filename specified for the specified `ParentProcessor` will be taken
- `Level` – The level of the processor within the hierarchy. The highest level, for the system as a whole, is set to "n" and subsequent sub-levels are defined as "n-1", "n-2", "n-3" and so on

The definition of the system hierarchy used to develop the ENBIOS prototype–which is, indeed, the only existing version at time of writing–was based on the system structure and available data from the Euro-Calliope energy model (Pfenninger and Pickering, 2018; Pickering et al., 2022; Tröndle et al., 2020). Further details of the development of this structure can be found in the literature (Martin et al., 2022). Note that structural processors are given for all electricity, heat and fuel production processes. Furthermore, as greenhouse gas (GHG) emissions are not included for the fuel produced and (presumably) used for transport processes–and, potentially, for any electricity and heat generation processes not otherwise included in the individual electricity and heat listings–a second set of fuel processes are also included to account for combustion processes. This also includes fuels produced via hydrogen-to-fuel processes. These are not LCA-related in any way but are included purely to account for additional GHG emissions. Emissions are calculated in accordance with known emissions factors defined within parameter definition sheets (see Section 3.2.4).

3.2.2 Structural process definitions

While the overall structure of the energy system is defined within the "`BareProcessors top levels`" sheet (see Section 3.2.1), a greater level of detail is required for structural processors. This is done within the "`BareProcessors simulation`" sheet. Here, the data is arranged in a similar fashion to the "`BareProcessors top levels`" sheet, but additional specification information is also included. The most important function of this sheet is to define the way that data is assigned and imported from external data sources. Again, the modelled data used in the development of the ENBIOS module is taken from the Euro-Calliope model developed for the European energy system within the SENTINEL project (SENTINEL, n.d.). The model outputs from Euro-Calliope consist of a series of around 30 .csv files, each of which report a different aspect of the modelled system. Although these files vary slightly between revisions, these .csv files typically contain columns that define the `storyline` or `scenario` being simulated, the energy source or technology being used (`techs`), the location that the data applies to (`locs`), the energy carrier (`carriers`) and the units that apply to the given value (`unit`). Sector (`sector`) and sub-sector (`subsector`) definitions also apply to energy consumption data. The actual values that apply to each row of data is given in the final column which is typically named after the file itself (i.e., the column containing values is named `flow_out_sum` within flow_out_sum.csv files).

With this in mind, the full list of columns that define the structural processors within the "`BareProcessors simulation`" sheet is as follows:

- `Processor` – This is ***<u>different</u>*** to the processor name within the "`BareProcessors top levels`" sheet. Here, the processor name refers to the "`techs`" field within the .csv file that this row gets data from
- `@SimulationCarrier` – The "`carriers`" field within the .csv file that this row gets data from

- `@SimulationVariable` – The filename/data column that this row gets data from
- `ParentProcessor` – The name of the structural processor from the "`BareProcessors top levels`" sheet that this row is associated with. That is, multiple data inputs from .csv files can be applied to the same structural processor. For example, two types of heat from natural gas (e.g., combined heat and power and heat only processes) could be applied to a common "heat from natural gas" structural processor
- `@EcoinventCarrierName` – The energy carrier that applies to the subsequent ENBIOS calculations. This is included because several different carrier types within the input data could be simplified to a common carrier type in ENBIOS. For example, diesel, kerosene and methanol derived from biomass could be simplified to a single carrier named "biodiesel" in ENBIOS
- `SubsystemType` – As in the "`BareProcessors top levels`" sheet. Typically "Local"
- `FunctionalOrStructural` – As in the "`BareProcessors top levels`" sheet. Should all be "Structural" in this sheet
- `Accounted` – As in the "`BareProcessors top levels`" sheet. However, in this sheet these will typically be "No", meaning they are only used for reference and not for accounting
- `@EcoinventFilename` – The name of the Ecoinvent .spold file that defines the LCI data at this processor
- `@SimulationScalingFactor` – The factor to apply if any scaling is required of input values. For example, you may wish to only use a certain percentage of an input for some reason, or split values for one input across two processors. Values can be entered manually (e.g., enter 0.62 to always assign 62% of the incoming value). However, rules can also be assigned which allow the user to apply different scaling to different data coming in from input files. The available parameters are hard-wired into ENBIOS: "time", "scenario", "region", "carrier", "tech". So, you must use these–and no others–in your formula.

  Expressions can be entered into the "@SimulationScalingFactor" field using a simple nomenclature that uses two possible sets of conditions followed by a third value which acts as the "else" value (i.e., for anything that does not satisfy the previous conditions). So, for example, for a situation where two combinations of "scenario" and "time" require two specific factors while all others are to be assigned to 50%, the required entry would be:

  ```
  ? scenario=="government_directed" and time==2015 -> 0.93,
  scenario=="market_driven" and time=="2030" -> 0.8, 0.5 ?
  ```

  Furthermore, it is possible to create additional layers within entries to cover more combinations of conditions. For example, for two "scenario" values, each having two "time" values, the following entry could be used:

  ```
  ? scenario=="government_directed" -> (?time==2015 -> 0.93,
  time==2030 -> 0.91, 0.9?), scenario=="market_driven" -> (?time==2015
  -> 0.87, time==2030 -> 0.86?), 0.5 ?
  ```

- `@SimulationToEcoinventFactor` – The factor to apply to the imported data to make the units align with those used in the Ecoinvent .spold file inventories. Technically, this operates in precisely the same way as `@SimulationScalingFactor`, but the two are kept separate as they are used for different purposes (i.e., this factor is used solely for converting units). For example, if the data coming from Euro-Calliope are given in TWh and the Ecoinvent inventory is for a MJ, a conversion factor of 3.6 x 10^9 needs to be applied so that the incoming data is converted to MJ. For fuels, where Ecoinvent inventories are often provided per kg of fuel, a calorific value (i.e., energy per kg fuel) from known sources will need to also be applied. Great care should be taken to ensure that these factors are entered accurately
- `@IAMCCode` – As the SENTINEL project has indicated that it may wish to report results according to the energy system nomenclature defined by the Integrated Assessment Model Consortium (IAMC) (IAMC, n.d.), this column defined the IAMC code that applied to this row

Note that the use of files with more than one `Processor` column (or more than one column at this level in the .csv file) is not currently possible For example, see the "final_consumption.csv" files sometimes being provided by Calliope, which has a "root" column for `sector` and a sub-column that defines `subsector` at a lower level. If you require values from `subsector` to be imported, the best solution would be to manipulate data within the "final_consumption.csv" file such that data in the `Processor` column contains a composite string that includes both aspects (e.g., for a row with a `sector` string of "Buildings" and a `subsector` string of "Heat", you could manually create a `sector` value of "Buildings_Heat" (or similar) in Excel and resave the file.

Use of system definition data from other sources

Although ENBIOS has been primarily developed to interface with outputs from the Euro-Calliope models used in SENTINEL, other input data can be used. For example, you may have a tabular series of system definition data derived from a government policy document or other modelling approach. Here, you will be required to convert the data into a format that mimics Calliope outputs, namely the so-called "friendly_data" definitions used within the SENTINEL project (Ali and Pfenninger, n.d.).

Compatible files are created using a file in the same format as a "standard" Calliope output file (e.g., see "flow_out_sum.csv" or "nameplate_capacity.csv"), with the required data entered and formatted as required.

3.2.3 Interface definitions

The list of interface types used within the ENBIOS calculations are defined here. These are defined in a similar fashion to the structural and functional processors in that a simple hierarchy is defined using the InterfaceType and ParentInterfaceType columns. The listed values in InterfaceType should match the data listed in the @EcoinventCarrierName column of BareProcessors simulation.

It is important to note that many "interfaces" are also derived for all of the individual inventory items defined within the imported .spold files. However, ENBIOS generates these automatically when it processes the inventory data (see Section 3.1.1), so it is not necessary to define these again in this sheet.

3.2.4 Parameter data definitions

Any parameters to be used in within customised calculations in ENBIOS can be specified in additional sheets of the "base" workbook file. Theoretically, any number of sheets and parameter values can be specified. The columns used to define parameters in such sheets is as follows:

- Parameter – Name of the parameter
- Type – Type of parameter, always "number"
- Domain – (not typically used, leave blank)
- Value – Numerical value
- Group – String that defines group that parameter belongs to (for information only)
- Description – String to describe parameter (for information only)

A selection of values defined in the current version of ENBIOS are listed below.

Raw materials

Defined in the "Parameters raw materials" sheet. Four sets of factors relating to critical raw materials (CRMs) are defined using data from the literature (Bobba et al., 2020; European Commission, 2020; Wendling et al., 2020). Values for the four factors–total EU consumption, materials supply risk, end-of-life recycling input rate (EoLRIR) and local impact score–are listed for the 55 CRMs defined by the European Commission that also have their inventories listed within Ecoinvent .spold files. These values are used to calculate three CRM-related indicators in accordance with known methodologies (Martin et al., n.d.; Talens-Peiró et al., 2022).

Combustion

Defined in the "Parameters combustion" sheet. Again, as GHG emissions are not included for fuel use that occurs outside of the electricity and heat processes, a second set of fuel processes are also included to account for combustion processes. The combustion emission factors to be applied at these processors are listed within this sheet, as defined in the literature (IPCC, 2021).

Employment hours

Defined in the "`Parameters employment`" sheet. In order to calculate the hours of human labour relating to the operation of a given capacity of heat and electricity production infrastructure or the production of a given mass of fuels, known values from literature (Ram et al., 2020; Rutovitz et al., 2015)–in hours per TW and hours per TWh–are listed.

3.2.5 Labour calculation definitions

Although the parameters that link infrastructure capacity and fuel production to hours of human labour can be defined in the "`Parameters employment`" sheet (see Section 3.2.4), the ways in which these values are used to calculate actual hours of human labour (or "human activity" in the nomenclature of MuSIASEM and NIS) must be explicitly defined for each processor. This is done in the "`Interfaces employment`" sheet.

The sheet contains the following columns:

- `Processor` – Defines which processor that the calculation applies to. Will be in the form of `ParentProcessor.Processor`, from the "`BareProcessors simulation`" sheet. For example, for a processor defined by a `ParentProcessor` of "Electricity_onshorewind" and `Processor` of "wind_onshore", enter "Electricity_onshorewind.wind_onshore"
- `InterfaceType` – Should always be set to "Labour"
- `Orientation` – Defines direction of labour. Can be "Input" or "Output". Assume generally set to "Input"
- `Value` – Defines the formula that applies for calculating labour values. Any parameters can be used in the formulae, although it is suggested that labour constants (see "Employment hours" section) from the "`Parameters employment`" sheet are used rather than typing them into each cell in this sheet. Note that alternative values of files specified as `@SimulationVariable` from the "`BareProcessors simulation`" sheet can also be used. That is, data can be imported from other input files or columns (i.e., other `@SimulationVariable` specifications), but everything else the same. A good example is something like installed capacity, where ENBIOS could take the energy mix data for the general system calculations, then look up the capacity definition *only* for the labour calculations. In this case, the formula for, say, onshore wind would be "EmpCap_WindOnshore * nameplate_capacity" meaning that the employment constant from the "`Parameters employment`" sheet – "EmpCap_WindOnshore" – will be multiplied by the corresponding value of "nameplate_capacity" in the input files, even though this value is not otherwise part of the calculations that occur at processors
- `Time` – Defines the timestep that the calculation applies to. Assume generally set to "Year"

3.3 Indicator definitions

Once all of the data inputs and other required parameters have been defined, the final set of inputs are those that define the indicators that represent the final outputs from the ENBIOS simulation. Two types of indicators are possible. Firstly, this involves defining the primary LCIA indicators that can be derived using LCIA methods from the database. A second set of customised indicators can be derived that use combinations of these, LCI inventory values and the other parameters defined in Sections 3.2.4 and 3.2.5. All of these are defined within the "`ScalarIndicators`" sheet. The sheet contains the following columns:

- `IndicatorsGroup` – Defines the group type of the indicator (e.g., LCIA, GHG, materials, etc.)
- `Indicator` – Indicator name
- `Local` – Determines if indicator applies to local or entire system. Three types of scalar indicator can be used: "Local", "System" and "Global". Note that "System" and "Global" are the same for current SENTINEL runs but could be used later if multi-system models are ever used
- `Processors` – Allows user to restrict the indicator to certain processors. This is useful, for example, for the combustion calculations which only apply to the combustion processors. The nomenclature to use in this case would be:

  ```
  //processor[starts-with(lower-case(@name),'fuelcombustion_')]
  ```

  This would only apply the indicator calculation to processors with names that start with "fuelcombustion_".
  This command uses XPath nomenclature. Should you need to create specific filters, various online resources are available; the Wikipedia entry is possibly a good place to start.
- `Formula` – The formula to apply to calculate the given indicator
- `Unit` – The units that apply to this indicator
- `AccountNA` – Accounts for "not available" statistics.
  - "N" – All variables implied in an indicator
  - "Nav" – Stated in the interface, with no quantity available
  - "Nap" – A not stated interface (i.e., does not apply to the processor)
- `UnitLabel` – The string label to apply when reporting units

The two indicator types are discussed in more detail below.

3.3.1 LCIA indicators

Any of the LCIA methods defined within the `lcia_implementation_to_csv` function (see Section 3.1.2) can be used to provide indicator outputs from the final simulations. Note that a list of available LCIA indicators can be obtained using the `list_lcia_indicators` function. This method allows the use of some flags to specify the level of resolution of the available LCIA indicators:

- --compartment: Include compartment in the generated LCIAMethod calls.

- --subcompartment: Include subcompartment in the generated LCIAMethod calls.
- --horizon: Include horizon in the generated LCIAMethod calls.

Usage:

```
enbios list_lcia_indicators nis_base_url lcia_indicators_list_file
```

Examples:

- Without flags:

```
enbios list_lcia_indicators
https://docs.google.com/spreadsheets/d/1Kkrbd-
GkaLtrJwH2xTFPoe217QqsZaiSfuFss-iUnsg/edit#gid=387453521
"C:\ENBIOS\lcia_implementation_to_csv\lcia_indicators_list.xlsx"
```

- To include the methods compartments and subcompartments:

```
enbios list_lcia_indicators
https://docs.google.com/spreadsheets/d/1Kkrbd-
GkaLtrJwH2xTFPoe217QqsZaiSfuFss-iUnsg/edit#gid=387453521
"C:\ENBIOS\lcia_implementation_to_csv\lcia_indicators_list.xlsx"
--compartment --subcompartment
```

Details of the required method can then be copied and pasted directly from this file into the `Formula` column of the “`ScalarIndicators`” sheet.

3.3.2 Custom indicators

Any other customised formulae can also be entered that can, theoretically, use any parameter or value that exists within the ENBIOS simulation. These are entered into the “Formula“ column of the “ScalarIndicators” sheet. Most basic mathematical formats can be used (e.g., similar to those used in Excel).

When Local is equal to “System” and “Global”, aggregate functions (SUM, AVG, MIN, MAX, COUNT) can also be used. For interfaces that have both input and output values, suffixes of “_input” and “_output” can be used (e.g., “Diesel_output * 0.5”). Examples of existing formulae can be observed in working “base” workbook files such as this one.

3.4 Summary

The required inputs to ENBIOS, detailed in the above sections, are summarised in Table 1.

Table 1 Summary of ENBIOS inputs

Feature or command	Description	Source	Comment
LIFE CYCLE ASSESSMENT			
LCI inventory data	List of all LCI data to be applied at structural processors	Output from `lci_to_nis`. Data obtained from all Ecoinvent .spold files defined in `@EcoinventFilename` column in "`BareProcessors`" simulation sheet	All `InterfaceType` values in output file are automatically assigned as interface types, in addition to those defined manually in "`InterfaceTypes`" sheet
LCIA methods	Definition of characterization factors to apply to every LCI item in Ecoinvent database, for every LCIA method	Output from lcia_implementation_to_csv	Should only be needed when new definitions released by Ecoinvent (approximately annually)
"BASE" WORKBOOK FILE			
`Links with Teams`	Basic description. Not functionally important	Manual	
`RefProvenance`	Provenance references. Not	Manual	Useful to annotate the sources used in interface values

	functionally important		
`RefGeographic`	Geographic references. Not functionally important?	Manual	Useful to annotate location of processors using NUTS codes
`Parameters raw materials`	Material-related parameters	Manual	Used for calculating critical raw material indicators
`Parameters combustion`	GHG emission factors (e.g., $kgCO_2$-eq/TWh) for different fuel types	Manual	Used for calculating GHG emissions for fuel combustion processes
`Parameters employment`	Hours of human labour per unit of capacity (heat and electricity) and unit of energy (fuel)	Manual	Used for calculating metabolic indicators
`DatasetDef`	Defines the LCIA methods dataset (used by `LCIAMethods` command)	output from lcia_implementation_to_csv	
`InterfaceTypes`	Hierarchical list of interface types	Manual	Types of biophysical flows between processors. Typically electricity, heat and different fuel types
`BareProcessors top levels`	Definition of energy system structure	Manually transcribed from graphical dendrogram	Hierarchical schema of the energy sector using MuSIASEM level definitions (structural and functional processors)

`BareProcessors simulation`	**Definition of information at structural processors**	**Parent column should be aligned to the structural processors in “`BareProcessors top levels`” sheet. `Processor`, `@SimulationCarrier` and `@SimulationVariable` columns link to columns in input data files.**	**Provides integration between external model results, LCA processes and ENBIOS structural processors**
`Interfaces employment`	**Definition of human labour calculations that apply at each structural processor**	**Manual**	**Typically use labour parameters from `Parameters employment` and the energy or power capacity specification to use with these parameters**
`ImportCommands LCI structural`	**Definition of LCI database location**	**Online location of workbook generated by `lci_to_nis`**	**A database of LCI activities used to provide raw LCI information to individual structural processors**
`LCIAMethods`	**General definition of all LCIA methods**		**Assimilates an external dataset, currently the output file from `lcia_implementation_to_csv`. Probably best to leave as is**
`ScalarIndicators`	**Definition of LCIA methods and custom indicators**	**LCIA method indicators defined using rows in output file from `list_lcia_indicators` function. Custom indicators are manually entered and can use any parameter within ENBIOS**	**Indicators to be calculated at each structural processor**
`MatrixIndicators`	**Not functionally important**		

4 Execution of simulations

4.1 Exploration of simulation data

The final simulation should now be ready to perform. However, an additional (optional) action has been included to allow the user to analyse the data prior to (or, indeed, after) the simulation. The action is known as `sentinel_to_nis_preparatory` and may be useful for exploring the input parameters or for troubleshooting purposes. The required parameters for this action are as follows:

- `sentinel_data_package_json_path` – The path to the index .json file that summarises the simulation results. In the test cases provided from Euro-Calliope, such files are provided alongside the various result files and use the so-called "friendly_data" definitions that follow a common nomenclature for elements of the energy system to be used by all partners within the project
- `nis_file` – The local path where the output report workbook will be generated

Usage:

```
enbios sentinel_to_nis_preparatory sentinel_data_package_json_path
nis_file
```

Example:

```
enbios sentinel_to_nis_preparatory C:\ENBIOS\
sentinel_to_nis_preparatory\scenario_a\datapackage.json
C:\ENBIOS\sentinel_to_nis_preparatory\scenario_a\nis_file.xlsx
```

The output .xlsx file will contain 11 sheets that define every aspect of the input files described in the .json file.

4.2 Simulation

Once all inputs have been prepared (see Section 3), the final simulation can be performed using the `enviro` action. The steps carried out by this action are as follows:

- Read in all input data
- Split the simulation into separately processable pieces. Fragmentation is performed by "Scenario" and "Region". Different times/years could be performed within the same simulation in future releases, but is currently untested
- For each simulation fragment, a fragment-model (NIS file) is generated:
    - For each technology:
        - Gather interfaces coming from the simulation and create a processor with those interfaces
        - Find the main output of the technology
        - Find an LCI processor for each technology and copy the interfaces, making them relative to the simulation output instead of the LCI reference output (assume they are the same)
        - Find a MuSIASEM parent for the technology and assign it to that parent
    - Generate a NIS model file
    - Submit this dynamically generated NIS model, and calculate LCIA and other indicators
    - Collect and accumulate results

Figure 2 presents a simplified flowchart for this procedure. The "Simulation output" box represents the data package of outputs from other SENTINEL models that act as the changing energy system inputs to be considered in ENBIOS. This data then enters the "fragmenter" process, to produce accountable fragments in parallel. Each fragment then enters an "assembler" process–considered to be the core of ENBIOS–which generates a model incorporating and integrating information from the MuSIASEM dendrogram and information extracted from the LCI files. The assembled model is then injected into a process called "NIS solver" which propagates available quantitative information and calculates indicators arranging them in the output dataset.

In Figure 2, the salmon-coloured pieces of information come from the "base" workbook. The green, hexagonal processes are undertaken within the ENBIOS and NEXINFOSYS (NIS) packages. The blue output dataset at the bottom of the figure has similar dimensions to those from the SENTINEL input data package adding indicators instead of measures to technologies and incorporating the processors from the MuSIASEM dendrogram.

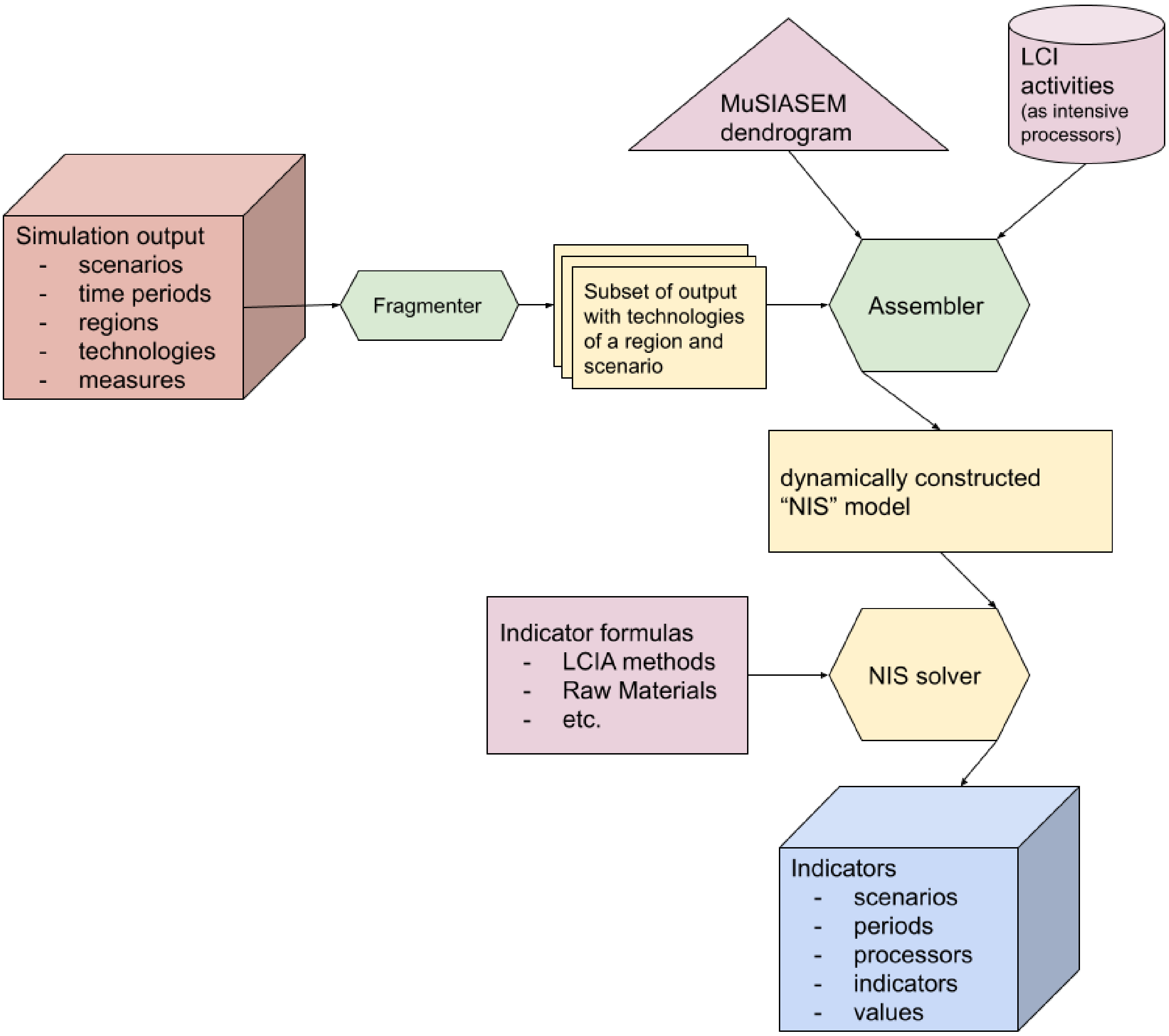

Figure 2 Conceptual flowchart for ENBIOS simulations

The `enviro` action requires the following parameter to be specified:

- `yaml_filename` – The path of the .yaml file that specifies the simulation (see below)

  The specified .yaml file contains the following lines:

  - `nis_file_location` – The URL of the "base" workbook file
  - `development_nis_file_location` – The URL of workbook used for testing. Not currently used

- `correspondence_files_path` – Not currently used. Could be used to define the path of a .csv file in which a mapping from technologies to .spold files and from technologies to the MuSIASEM dendrogram could be specified
- `simulation_type` – States the file format of the simulation input. It should be set to "sentinel" when input simulation data is taken from Calliope or Euro-Calliope. It should be set to "single_csv" when input simulation data is manually defined in a csv file.
- `simulation_files_path` – Path to the reference file or folder needed by the specific simulation output format. In the case of a "sentinel" simulation, this should be the path of the index .json file, as used in the sentinel_to_nis_preparatory function
- `output_directory` – Path of the local directory where all outputs of the execution will be placed

An example of a typical .yaml input file is given below:


```
nis_file_location: "https://docs.google.com/spreadsheets/d/1Kkrbd-GkaLtrJwH2xTFPoe217QqsZaiSfuFss-iUnsg/edit?usp=sharing"
development_nis_file_location: ""
correspondence_files_path: ""
simulation_type: sentinel
simulation_files_path:
D:/ENBIOS/sentinel_to_nis_preparatory/datapackage.json
output_directory: output
```


A number of "flags" can also be specified after the `yaml_filename` specification as part of an `enviro` command:

- `--n-fragments` – Calculates the first "n" fragments (i.e., regions). Low numbers can reduce run time for testing
- `--first-fragment` – If a specific fragment (i.e., region) is required, the number of that fragment can be specified
- `--fragments-list-file` – Generates a file "fragments_list.csv" that lists the expected fragments for the given inputs. This is useful if a --first-fragment flag is to be used (i.e., this flag will provide a list of fragments and their number so that the correct fragment can be selected)
- `--keep-min-fragment-files` – To keep a copy of the NIS file submitted to NIS
- `--generate-interface-results` – Generates a .csv file containing the values at each interface for each fragment
- `--generate_indicators` – If "true", a separate "indicators.csv" file (the main results file type – see Section 5) is written for each of the fragments (the aggregate "indicators .csv" file is always generated). "false" by default

- o `--max_lci_interfaces` – If >0, when LCI interfaces are merged into simulation processors, the number of interfaces cannot exceed this parameter. This is useful when checking the system prior to a full execution. By default, this is set to 0, meaning “all interfaces” are activated
- o `--generate_nis_base_file` – Because the “base” framework model can be quite large, it is possible to have specification issues. If this parameter is set to “true”, a workbook in NIS format is generated returning information on how NIS interprets “base”. “false” by default
- o `--generate_nis_fragment_file` – If “true”, for each fragment the dynamically generated NIS file is written to the output directory. “false” by default
- o `--n-cpus` – The number of CPUs to be used to process several fragments in parallel. Can be used to reduce simulation times on more powerful computers or clusters

Usage:

```
enbios enviro yaml_filename -n-fragments --first-fragment --
fragments-list-file --keep-min-fragment-files --generate-interface-
results --generate-indicators --max-lci-interfaces=1 --
generate_nis_base_file --generate_nis_fragment_file --n-cpus
```

Examples:

- For a short test using three fragments, starting at fragment #9...

```
enbios enviro C:\ENBIOS\enviro\scenario_a.yaml --first-fragment=9 --
n-fragments=3
```

- To determine fragment numbers (e.g., if you want to test a specific region)...

```
enbios enviro C:\ENBIOS\enviro\scenario_a.yaml --fragments-list-file
--n-fragments=1 --max-lci-interfaces=1
```

- For full simulation (i.e., all fragments) using four CPUs...

```
enbios enviro C:\ENBIOS\enviro\scenario_a.yaml --n-cpus=4
```

5 Results

5.1 Output files

The main output of the process is an “indicators.csv” file which details the indicator results for each fragment at each defined processor. The file contains the following columns:

- `Scenario` – To which scenario, coincident with SENTINEL scenario codes, do the indicator values apply
- `Period` – As above, but relating to time period
- `Scope` – Scope of the indicator. Can be “Internal” (domestic), “External” (imports/exports) or “Total” (combination of these two). Current simulations only consider “Internal” amounts, in which case “Internal” and “Total” will be the same
- `Processor` – The processors to which the indicator applies
- `Indicator` – Name of the indicator
- `Value` – Value of the indicator

Intermediate .xlsx files are also be returned for each fragment and scenario. These can be useful for troubleshooting and validation.

5.2 Visualizer

Because the “indicators.csv” files are often large and complicated, a visualizer has been derived that provides tabulated summaries of the results. The visualizer is opened in your default browser application (Chrome, etc.).

Usage:

```
enbios visualize yaml_filename
```

Then, to open the visualizer itself, go to http://localhost:8050

Example:

```
enbios visualize D:/ENBIOS/enviro/scenario_a.yaml
```

http://localhost:8050

6 References

Ali, S., Pfenninger, S., n.d. Friendly data.

Bobba, S., Carrara, S., Huisman, J., Mathieux, F., Pavel, C., 2020. Critical raw materials for strategic technologies and sectors in the EU: A foresight study. Publications Office of the European Union, Luxembourg. https://doi.org/10.2873/865242

di Felice, L.J., Ripa, M., Giampietro, M., 2019. An alternative to market-oriented energy models: Nexus patterns across hierarchical levels. Energy Policy 126, 431–443. https://doi.org/10.1016/j.enpol.2018.11.002

Ecoinvent, 2021. Ecoinvent version 3.8 (2021) database [WWW Document]. URL https://v38.ecoquery.ecoinvent.org/Home/Index

European Commission, 2020. Study on the EU's list of critical raw materials (2020) - Final report. Publications Office of the European Union, Luxembourg. https://doi.org/10.2873/11619

Giampietro, M., Mayumi, K., Ramos-Martin, J., 2009. Multi-scale integrated analysis of societal and ecosystem metabolism (MuSIASEM): Theoretical concepts and basic rationale. Energy 34, 313–322. https://doi.org/10.1016/j.energy.2008.07.020

González-López, R., Giampietro, M., 2018. Relational analysis of the oil and gas sector of Mexico: Implications for Mexico's energy reform. Energy 154, 403–414. https://doi.org/10.1016/j.energy.2018.04.134

Huijbregts, M.A.J., Steinmann, Z.J.N., Elshout, P.M.F., Stam, G., Verones, F., Vieira, M., Zijp, M., Hollander, A., van Zelm, R., 2017. ReCiPe2016: A harmonised life cycle impact assessment method at midpoint and endpoint level. International Journal of Life Cycle Assessment 22, 138–147. https://doi.org/10.1007/s11367-016-1246-y

IAMC, n.d. Integrated Assessment Modeling Consortium wiki [WWW Document]. URL https://www.iamcdocumentation.eu/index.php/IAMC_wiki

IPCC, 2021. Emission factor database [WWW Document]. URL https://www.ipcc-nggip.iges.or.jp/EFDB/main.php

MAGIC, n.d. MAGIC - NIS front-end how-to manual.

Martin, N., Madrid-López, C., Villalba-Méndez, G., Talens-Peiró, L., n.d. Optimising transition pathways. New techniques for assessing critical raw material constraints in energy and other technologies Manuscript.

Martin, N., Talens-Peiró, L., Villalba-Méndez, G., Nebot-Medina, R., Madrid-López, C., 2022. The energy transition beyond climate neutrality: Deeper evaluations of impacts and constraints in future systems Manuscript.

Nebot Medina, R., Martin, N., Talens-Peiró, L., Madrid-López, C., n.d. The ENvironmental and BIOeconomic System Assessment (ENBIOS) module.

Nebot Medina, R., Moreno Fajardo, P., Galluzzi, M., Staiano, M., 2020. Report on the material produced for the Nexus knowledge hub: Report on NIs toolset development. MAGIC (H2020–GA 689669) project, Annex to Deliverable 3.3.

Pfenninger, S., Pickering, B., 2018. Calliope: A multi-scale energy systems modelling framework. Journal of Open Source Software 3, 825. https://doi.org/10.21105/joss.00825

Pickering, B., Lombardi, F., Pfenninger, S., 2022. Diversity of options to reach carbon-neutrality across the entire European energy system. Joule 6, 1253–1276. https://doi.org/10.1016/j.joule.2022.05.009

Ram, M., Aghahosseini, A., Breyer, C., 2020. Job creation during the global energy transition towards 100% renewable power system by 2050. Technological Forecasting and Social Change 151, 119682. https://doi.org/10.1016/j.techfore.2019.06.008

Rutovitz, J., Dominish, E., Downes, J., 2015. Calculating global energy sector jobs: 2015 methodology update. Institute for Sustainable Futures, University of Technology Sydney, Sydney.

SENTINEL, n.d. Sustainable Energy Transitions Laboratory (SENTINEL) [WWW Document]. URL https://sentinel.energy/

Talens-Peiró, L., Martin, N., Villalba-Méndez, G., Madrid-López, C., 2022. Integration of raw materials indicators of energy technologies into energy system models. Applied Energy 307, 118150. https://doi.org/10.1016/j.apenergy.2021.118150

Tröndle, T., Lilliestam, J., Marelli, S., Pfenninger, S., 2020. Trade-offs between geographic scale, cost, and infrastructure requirements for fully renewable electricity in Europe. Joule 4, 1929–1948. https://doi.org/10.1016/j.joule.2020.07.018

Wendling, Z.A., Emerson, J.W., de Sherbinin, A., Esty, D.C., 2020. Environmental performance index 2020. Yale Center for Environmental Law & Policy, New Haven, CT.

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