Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original Software Publication

pyGRETA, pyCLARA, pyPRIMA: A pre-processing suite to generate flexible model regions for energy system models

Kais Siala [*], Leonhard Odersky

Chair of Renewable and Sustainable Energy Systems, Technical University of Munich, Germany

ARTICLE INFO

Article history:
Received 15 September 2020
Received in revised form 10 July 2021
Accepted 19 October 2021

Keywords:
Energy system modeling
Spatial complexity
Clustering
Pre-processing

ABSTRACT

This paper presents a combination of three pre-processing tools that allow energy system modelers to define the number and shape of their model regions flexibly. Firstly, weather reanalysis data and other geographic maps are combined in pyGRETA to downscale wind and solar data and obtain renewable energy potential maps in high spatial resolution, while pyPRIMA can provide the spatial distribution of the energy demand and a pre-processed network of transmission lines. Secondly, the raster maps and the transmission grid are fed into pyCLARA to obtain a shapefile of regions with homogeneous characteristics. Thirdly, the obtained shapefile is used in pyGRETA to generate representative time series of renewable power generation, and in pyPRIMA to pre-process the rest of the data (power plants, demand, grid, etc.) to prepare input files for model frameworks. The three tools have a similar software architecture and are available in GitHub with an open source license and a detailed description. A minimal working example shows how they can operate together to ensure a high degree of modeling flexibility.

Code metadata

Current code version	pyGRETA [1]: v1.1.0 pyCLARA [2]: v1.0.1 pyPRIMA [3]: v1.0.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_112
Code Ocean compute capsule	–
Legal Code License	GNU General Public License (GPL)
Code versioning system used	git
Software code languages, tools, and services used	python
Compilation requirements, operating environments & dependencies	Check `pyGRETA.yml`, `pyCLARA.yml`, and `pyPRIMA.yml` in the `env` folders in the respective repositories.
If available Link to developer documentation/manual	https://pygreta.readthedocs.io/ https://pyclara.readthedocs.io/ https://pyprima.readthedocs.io/
Support email for questions	Please use the "Issues" function in the GitHub repositories. kais.siala@tum.de

1. Motivation and significance

The decarbonization of energy systems is posing multiple challenges that need to be addressed in energy system models. First, the share of decentralized generation from renewable energy sources, particularly solar and wind, is increasing rapidly. However, this renewable generation fluctuates over time and space and tends to be located far from load centers. It is therefore necessary to reflect this spatial and temporal discrepancies in the models, which is only possible if the input data is provided in high resolution. Pfenninger et al. [4] identified the latter point as the first challenge for twenty-first century energy system modeling.

Second, system decarbonization goes beyond the power sector and concerns industry, transport, heating and cooling. Klein

[*] Correspondence to: Potsdam Institute for Climate Impact Research, Potsdam, Germany.
E-mail address: kais.siala@tum.de (Kais Siala).

https://doi.org/10.1016/j.softx.2021.100860

et al. [5] included industrial feedstock, heating, road and maritime transport, and aviation, and estimated that around 2000 TW h of renewable power is needed, beside biomethane and geothermal energy. From the spatial perspective, an expansion of this magnitude will require the use of 2% of the area of Germany for onshore wind and utility-scale solar photovoltaic projects according to Klein et al. [5]. This again amplifies the importance of modeling the potential of these technologies with a high spatial resolution, to identify the best available project sites and minimize their spatial footprint.

Sector coupling poses another challenge from the spatial perspective. The different energy sectors have characteristics which cannot be modeled at one spatial scale. For instance, modeling the transport sector can be at the scale of cities, countries, or the whole globe, depending on the mode of transportation. At each scale, there is a multitude of stakeholders. Thus, energy system modelers need more flexibility in designing their models to capture processes and flows at each spatial scale. Only with flexible spatial units that adapt to their scope can they represent the emerging trends in energy supply, as exemplified by the cellular approach of Benz et al. [6] and Amado et al. [7].

The trend towards models with a high spatial resolution and which are capable of representing multiple energy sectors at different scales is partly hampered by the unavailability of data and/or by limited computational power. Even when open data is available, it is usually at a low resolution and requires some processing before it can be used in energy system models. As of the computational power, the modeling of multiple energy sectors with high spatial and temporal resolutions under uncertainty might be challenging using regular office computers at the disposal of modelers. Hence, many studies have attempted to identify the smallest number of regions for which the errors that arise from the simplification are acceptable, either through clustering [8] or heuristics [9,10]. Frysztacki et al. [11] provide a valuable review of recent studies applying spatial aggregation and clustering, most of which describe their algorithms without providing the software to replicate their work.

Such studies that vary the model regions are the exception, not the rule. Despite the need for high resolution data and for model regions that adapt to the research questions, the overwhelming majority of energy system models still relies on administrative and political divisions of space. The European Commission relies on different models (e.g. JRC-EU-TIMES [12], METIS [13], POTEnCIA [14], JRC-GEM-E3 [15]), which, despite the diversity of their techniques and objectives, use countries or groups of countries as their spatial units. The open-source model framework OSeMOSYS [16] has a model version for the EU countries plus Norway and Switzerland (OSeMBE), beside other versions for South America (SAMBA) and Africa (TEMBA), which use countries as model regions.

Considering the challenges of renewable energy integration and sector coupling, there is a need for more flexibility in defining the scale and the shape of the spatial units, which should not be restricted to political and administrative divisions. The pre-processing suite made of the three open-source tools pyGRETA,[1] pyCLARA[2] and pyPRIMA[3] has been designed for that purpose. Unlike webtools and global atlases of high resolution data, it allows the modeler to also aggregate the data and restructure it as input for model frameworks. Such an aggregation can be data-driven (i.e. through clustering algorithms) or user-defined (i.e. using given shapefiles of custom model regions). Table 1 provides an overview of commonly used tools by energy system modelers, their similarities with the proposed pre-processing suite, and the additional features of the latter. While most basic features are covered by existing tools, the novelty of the pre-processing suite resides in its wide array of advanced features and its ability to operate on user-defined regions within any geographic scope.

The energy system model PyPSA-Eur [17] deserves particular mention because it includes modules for data acquisition and cleaning, time series generation, and clustering. It therefore shares a lot of similarities with the pre-processing suite as a whole. Whereas PyPSA-Eur has the advantage of a smooth usage since all the modules are within the same tool and the data acquisition is automated, the proposed pre-processing suite stands out for being model-agnostic and adaptable to any geographic scope (not only within Europe). The three tools have also some advanced features (e.g. more clustering options, ability to use user-defined regions) which energy system modelers can make use of for special applications.

We believe that the pre-processing suite has an added value in terms of reproducibility and transparency, because the open-source codes and their assumptions are documented and the generation of the model input files can be replicated. It is particularly suited for studies investigating the impact of the spatial resolution of energy system models, as conducted by Siala and Mahfouz [27].

In addition to the aforementioned study, which used the complete pre-processing suite, modules of pyGRETA and pyPRIMA have also been used in other publications by one of the authors [28,29]. pyCLARA is currently used in a project on the near-surface geothermal potential of a German city, by clustering parts of it based on the underground water temperature. pyGRETA is used in several studies on different regions of the world, in particular for the estimation of the hydrogen potential that can be obtained through wind-powered electrolysis in the Middle East.

2. Software description

The software is published in three separate repositories, because each tool can be operated independently. However, the scripts have the same coding style and architecture, so that it is easy to use them together. Section 2.1 describes the common architecture, followed by an overview of the functionalities of each tool in Section 2.2. Finally, Section 2.3 suggests a workflow that combines the tools of the pre-processing suite.

2.1. Software architecture

The pre-processing suite is composed of three independent repositories: pyGRETA, pyCLARA, and pyPRIMA.

Each repository includes the following items:

- Folder `code`: this includes the script `config.py` for user preferences, the script `runme.py` that calls the other modules, and the folder `lib` which contains the submodules.
- Folder `doc`: this includes all the files needed to generate the documentation in readthedocs.io.
- Folder `env`: this includes the YAML file with all the dependencies. Use it in `conda` to create an environment where you can run the code.
- `LICENSE` file.
- `README.md` file for an overview of the repository.

The repositories use the *Black*[4] coding style for harmonization and improved readability, and follow the *all-contributors*[5] specification to encourage participation in code development.

[1] **py**thon **G**enerator of **RE**newable **T**ime series and m**A**ps.

[2] **py**thon **C**lustering of **L**ines **A**nd **RA**sters.

[3] **py**thon **PR**eprocessing of **I**nputs for **M**odel fr**A**meworks.

[4] https://github.com/psf/black.

[5] https://github.com/all-contributors/all-contributors.

Table 1
Comparison with existing software tools.

Software	Closest match	Similarity	Advantages of the pre-processing suite
pyGRETA	atlite [18]	Calculation of potentials and generation of time series	Correction of reanalysis data through combination with land use and topography maps; potentials and time series calculation after increasing the resolution; multiple representative time series per region
	GLAES [19,20]	Modeling of land eligibility constraints	Calculation of the resource and technical potentials; generation of representative time series
	Global Atlas [21]	Calculation of potentials for points or custom regions	generation of representative time series
	Global Wind Atlas [22]	Calculation of the potentials	Modeling of land eligibility constraints; generation of representative time series
	Global Solar Atlas [23]	Calculation of the potentials	Modeling of land eligibility constraints; generation of representative time series
pyCLARA	PySAL [24]	Clustering of regions using max-p regions algorithm (library is used in pyCLARA)	Pre-setting and adaptation of the algorithm to the application on large-scale energy systems; additional custom algorithm for transmission lines
pyPRIMA	OPSD [25]	Different scripts for cleaning and visualizing raw data	Pre-processing scripts to prepare model input files in the model format (currently for urbs and evrys)
	pyam [26]	Modules for data wrangling; common data structure for different models	Modules and structure more adapted to regional energy system optimization models (as opposed to global integrated assessment models)

First time users are encouraged to read the documentation of each tool, which is made of three parts:

- User manual: with an installation guide, descriptions of `config.py` and `runme.py`, a list of recommended input sources, and a recommended workflow.
- Theory: this includes some theoretical background, which the code is based on.
- Implementation: this describes all the submodules of the script, i.e. what they are used for, which inputs they use and which outputs they deliver.

The software is written in such a way that the average user only needs to edit `config.py` and `runme.py`. The modules that are called in `runme.py` are run in sequence, and save their outputs locally, including a JSON file that documents which parameters and paths were used to run them. If an error occurs and the script is interrupted, the user can fix the issue and restart `runme.py` after commenting the steps that have been completed successfully. The only variables that are used in every module are the dictionaries `param` and `paths`, which are generated in the initialization function based on the user preferences in `config.py`.

2.2. Software functionalities

2.2.1. Renewable energy potentials and time series

The first tool, pyGRETA [1], allows the user to estimate the theoretical and/or technical potential of an area in high resolution, for various technologies (onshore wind, offshore wind, photovoltaics, and concentrated solar power). The strength lies in the flexibility of defining the technological characteristics (by setting their parameters) and the region of interest (by providing a shapefile). Currently, it uses MERRA-2 reanalysis data, with the option to detect and correct outliers. For the potential estimation, it takes into account land use suitability/availability, topography, bathymetry, slope, and distance to urban areas. There is an option to provide the results of the potential estimation as raster maps and/or statistical reports with summaries (available area, maximum capacity, maximum energy output, etc.) for each user-defined region. Using the potential maps, and after excluding unsuitable parts for renewable power plants, the user can generate multiple time series for each region of interest, for example for the best site, the one at the upper 10% quantile, the median, etc. It is also possible to combine the time series into one using linear regression to match historical full-load hours and temporal fluctuations.

2.2.2. Clustering of rasters and networks

The pyCLARA tool [2] takes as input high resolution data (as delivered by pyGRETA, for example), and clusters it to obtain regions for energy system models with homogeneous characteristics. There are two modes of operation. It either uses one or multiple rasters of the same size and geographical coverage, such as wind resource maps or load density maps, and clusters them simultaneously, provided that the user has defined the weighting factor for each input and its corresponding aggregation function (average, sum, or density), in addition to the target number of regions. The code uses a combination of k-means and max-p algorithms, to ensure computational speed and spatial contiguity of the results. The other option is to cluster a network, for example a transmission grid, using a hierarchical algorithm. Thus, the modeler would be able to take into account grid restrictions when defining regions for power system modeling. Here again, the final number of regions can be chosen freely.

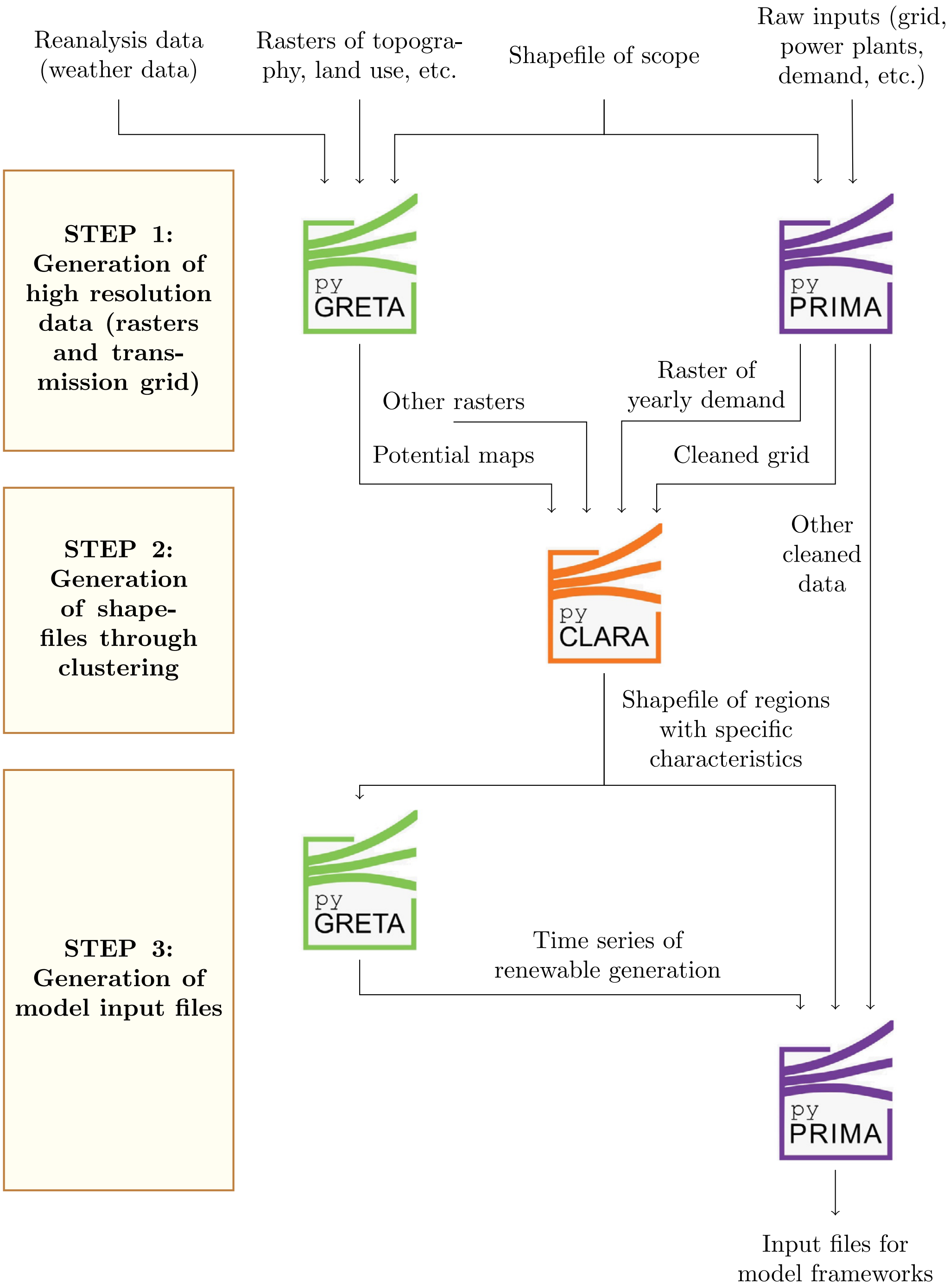
Reanalysis data (weather data)
Rasters of topography, land use, etc.
Shapefile of scope
Raw inputs (grid, power plants, demand, etc.)
STEP 1: Generation of high resolution data (rasters and transmission grid)
py GRETA
py PRIMA
Other rasters
Raster of yearly demand
Potential maps
Cleaned grid
STEP 2: Generation of shapefiles through clustering
py CLARA
Other cleaned data
Shapefile of regions with specific characteristics
STEP 3: Generation of model input files
py GRETA
Time series of renewable generation
py PRIMA
Input files for model frameworks

Fig. 1. Recommended workflow.

2.2.3. Flexible creation of model inputs

The pre-processing tool pyPRIMA [3] automates the creation of energy system models using a common database. The emphasis lies in the flexibility of defining the model regions and in the harmonization of the assumptions, which are well-documented and less prone to human errors due to automation. It operates in two steps. First, it reads raw input data for the spatial scope of the analysis and "cleans" it. This involves the filtering and renaming of entries and attributes, the correction of erroneous data points, the completion of missing data points using assumptions, the conversion of units, and the restructuring of the resulting database such that it has a clear, model-independent format. Currently, the tool includes modules for reading raw input data for Europe, but it can be expanded to include other data sources for other regions, provided that the output format is preserved. Second, the modeler chooses the shapefile of the model regions to use, and the tool converts the model-independent intermediate files into input files that can be used by the model frameworks. In the current version, two model frameworks, *urbs*[6] and *evrys*[7], are supported. If needed, new modules can be added easily to cater for other energy system model frameworks.

[6] https://github.com/tum-ens/urbs.

[7] Internal, closed-source tool for unit commitment optimization used at TUM, Germany.

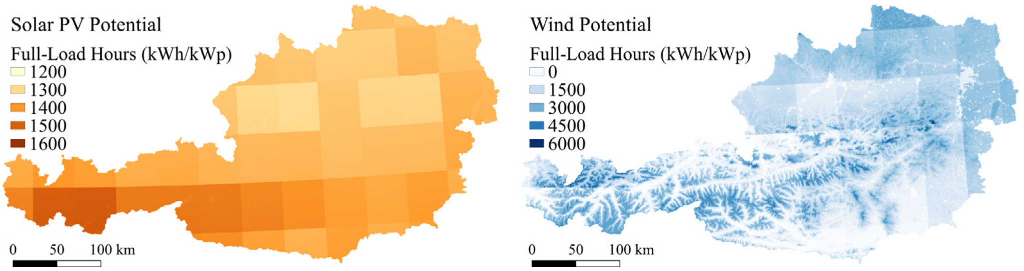

Fig. 2. Solar photovoltaic (left) and onshore wind (right) potentials for Austria according to pyGRETA.

2.3. Recommended workflow

Fig. 1 presents a generalized workflow that combines the three tools pyGRETA, pyCLARA and pyPRIMA. As stated before, the tools can be operated independently, although it is their combination that allows the modelers to flexibly generate the input files for their energy system models.

Some steps can be skipped in the following cases:

- If the regions are pre-defined and no clustering is needed, then *step 2* can be skipped.
- If the time series are available from another source, then the use of pyGRETA in *step 1* and *step 3* can be skipped.

3. Illustrative examples

In order to demonstrate the major functionalities of the pre-processing suite, we used it to create input files for an energy model of Austria. The model regions are based on the homogeneity of the PV and wind potentials. The next paragraphs are a step-by-step guide on how to replicate this example.

Before you start

The code and the data sets generated by the tools are published altogether [30]. The raw input files are not included in the repository. However, they can be downloaded from their respective sources, as mentioned in the documentations of the tools.[8] After downloading and unpacking the tools, make sure that the raw input data is saved in the corresponding folder in the database, so that the codes can run smoothly. If you have issues obtaining the raw input data, you can test the subsequent modules of the tools using the intermediate files that are shared.

Step 1a: Potential maps with pyGRETA

Firstly, pyGRETA generates the potential maps of PV and wind for Austria. Ensure that the used shapefile of Austria has a column with the name `NAME_SHORT`, eventually by adding it manually. The shapefile has only one feature, which will be referred to as `AUT`. Also the user should adapt parameters and file paths in the `config.py` which can be found in the sub-folder *code* of the pyGRETA folder. To run the code the script `runme_step1 A.py` is executed, which is place in the same sub-folder. The results can be seen in Fig. 2.

Step 1b: Pre-processed data for the whole scope

Secondly, pyPRIMA is used for cleaning the raw input data. As in the first step, the `config.py`, which is placed in the subfolder *code* of pyPRIMA, might be adapted to the correct input paths. Use `runme_step1B.py`, which can be found in the same subfolder, to execute the code of pyPRIMA. Some results regarding the load time series and the spatial distribution of power plants are shown in Fig. 3.

Step 2: Model regions through clustering

Thirdly, the regions for the energy model are defined using pyCLARA. For this example, the PV and wind potential maps generated with pyGRETA are used. To run pyCLARA, the file `runme_step2.py` in the subfolder *code* of pyCLARA is executed. The results of the clustering algorithm can be seen in Fig. 4. Since the code is not entirely deterministic, size, shape and number of clusters can vary from one run to another.

Step 3a: Time series for model regions

After defining the model regions with the clusters of pyCLARA, it is time to prepare the inputs for each of these regions. Therefore, pyGRETA is used again to create the time series of renewable generation. Since no regression is possible for the custom regions, two files at the location `Database\ 03 Intermediate files\ Files Austria\ Renewable energy\ Regional analysis\ Austria\ Regression outputs` were added manually. If pyCLARA generated a different number of clusters, ensure that the number of columns and the column names in these files match the number and names of the clusters generated in the previous step. There are no other changes necessary in `config.py`, so you can directly execute the script `runme_step3 A.py`, which is again placed in the subfolder `code` of the pyGRETA folder. A portion of the time series for clusters CL01 and CL04 are plotted in Fig. 5.

Step 3b: Model input files

Lastly, pyPRIMA is used to create the input file for the energy system framework *urbs*. For this purpose, the script `runme_step3B.py`, which can be found in the subfolder `code` of pyPRIMA, is executed. Fig. 6 shows some of the pre-processed inputs of the model.

4. Impact

This pre-processing suite made of three open-source tools covers all the steps needed to create energy system models with a flexible set of model regions. Whereas there are other tools in the modeling community with partially similar capabilities (see Table 1), this is the first attempt to automate the process of generating energy system models for custom regions within any

[8] Please check the additional notes in the Zenodo repository for links to the data sources.

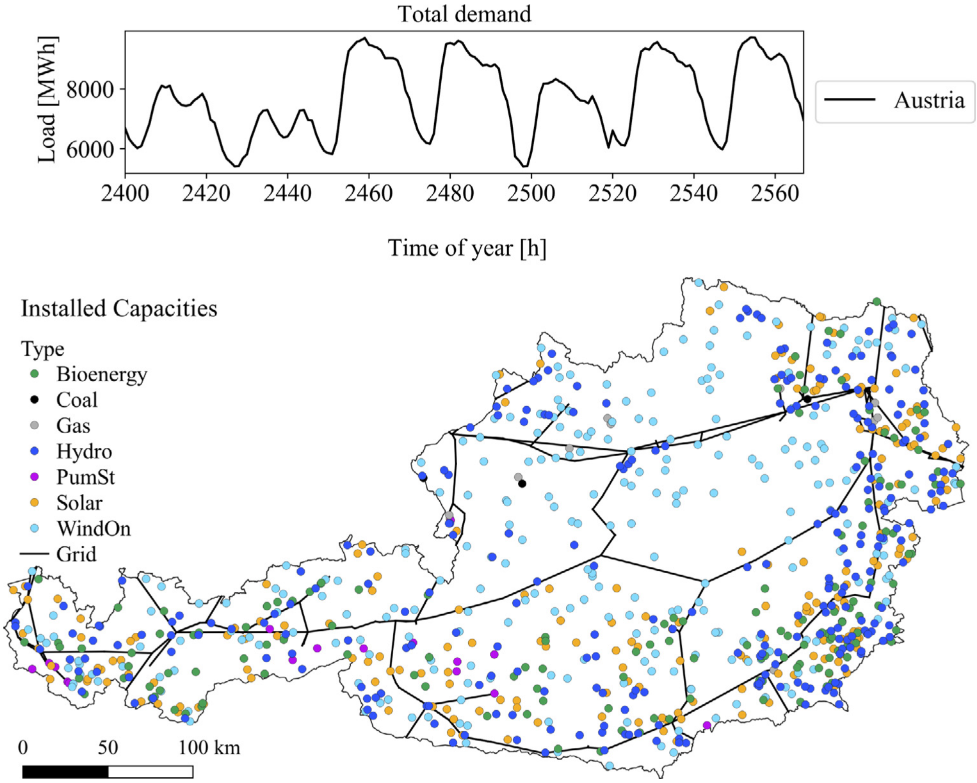

Fig. 3. Hourly load time series of Austria for a selected week (top) and spatial distribution of power plants and main transmission lines (bottom) according to pyPRIMA.

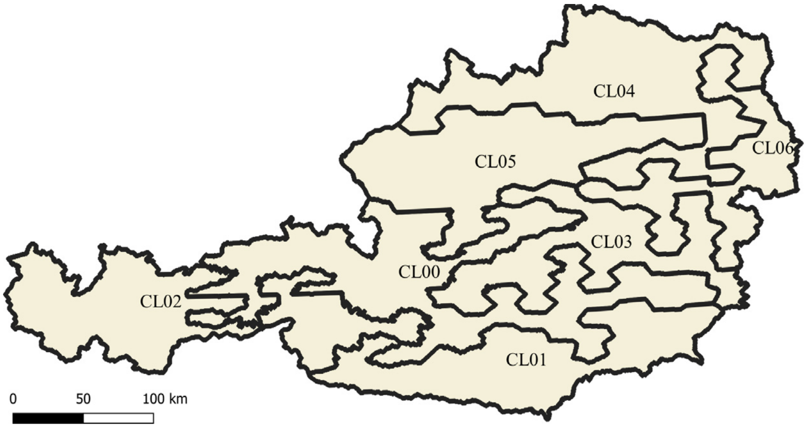

Fig. 4. Model regions obtained after clustering the potential maps using pyCLARA.

geographic scope using tools with similar software architecture. The availability of the code as open-source with an extensive documentation means that the suite can be easily adopted by prospective modelers.

There are two main usage cases for the pre-processing suite as a whole. First, it provides an essential toolset for modelers investigating the effect of the spatial resolution on model results. Similarly to Siala and Mahfouz [27], it allows the modelers to define the model regions based on certain characteristics and analyze the impact thereof. A large number of energy system models can be generated automatically to study the relevance of the spatial resolution systematically, transparently, and reproducibly. Such empirical experiments can challenge the preconception that administrative divisions are suitable for most research questions. Second, it facilitates the model creation with customized regions if that is dictated by the research question. This is for example the case when the geographic scope covers an area that does not overlap with administrative divisions for which the data is available. Another example is when certain aspects, such as grid bottlenecks, are so critical for answering the research question adequately, that they take precedence over administrative boundaries and require the usage of unconventional model regions.

The three tools include some novel features that offer new possibilities for energy system modelers. For instance, modelers can set up custom technology characteristics for renewable technologies in pyGRETA and generate time series using them. This is particularly useful when modeling future scenarios with technologies that do not exist yet. The tool pyCLARA accepts a wide set of inputs which do not have to be energy-related, provided that the rasters are of the same size and that their

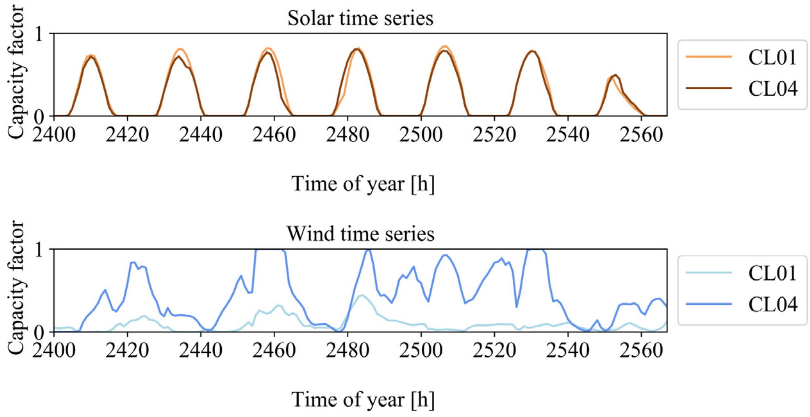

Fig. 5. Solar photovoltaic (top) and onshore wind (bottom) time series for the regions CL01 and CL04 for a selected week according to pyGRETA.

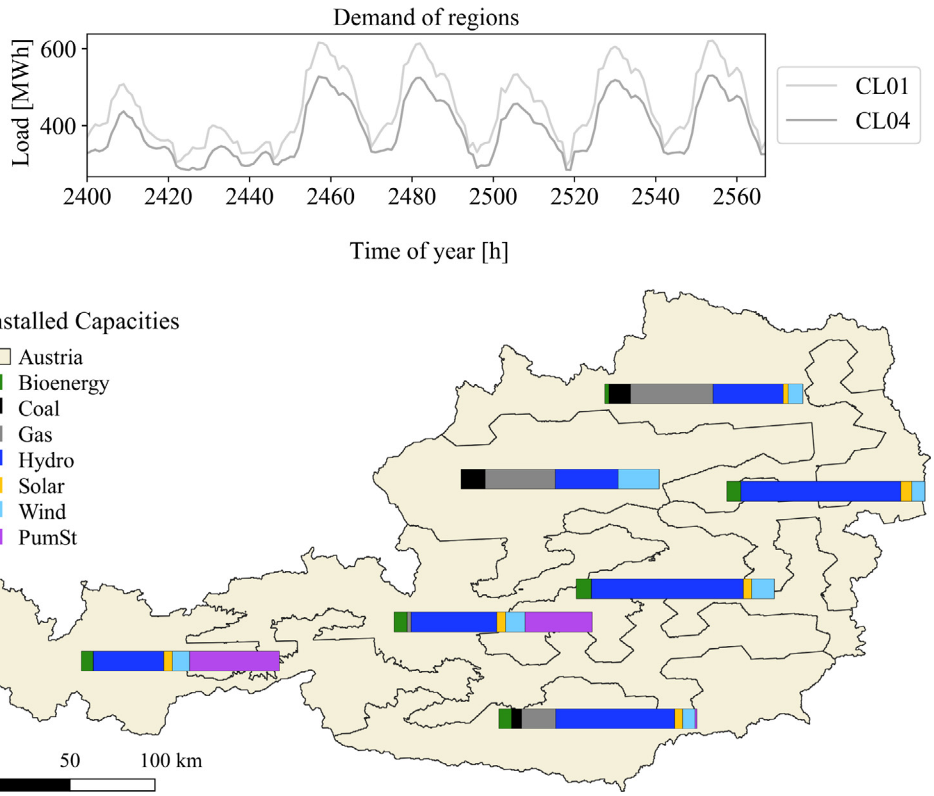

Fig. 6. Hourly load time series for the regions CL01 and CL04 (top) and share of technologies in the installed capacities of each region (bottom) according to pyPRIMA.

data can be aggregated by calculating the sum, the average or the density. Furthermore, since pyPRIMA takes into account time-dependent parameters (cost assumptions, construction year, etc.), it can create input files for different years, extending therefore the possibilities across the temporal dimension.

The three tools are currently used in different projects at the Chair of Renewable and Sustainable Energy Systems, TUM (Germany). Uptake in the wider modeling community has been growing steadily since their publication in June 2020, as inferred from the requests for assistance in GitHub or through emails. However, there is currently no mechanism in place to track the actual usage of the software.

5. Conclusions

The pre-processing suite presented in this paper is made of three independent tools that generate data in high spatial resolution, aggregate it according to pre-defined model regions or data-driven clusters, and prepare all the necessary input in the appropriate format for energy system modeling frameworks. The suite opens up new possibilities to modelers who are eager to investigate the impact of the choice of the spatial resolution, or who are compelled to use unconventional model regions to answer their research questions adequately. An illustrative example for Austria showed some of the capabilities of the tools. A detailed documentation is available in each GitHub repository, where the tools are shared with open-source licenses to encourage widespread use and participation in their development.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author acknowledges the financial support by the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) in the project "4NEMO - Research Network for the Development of New Methods in Energy System Modeling; research project: 0324008A". They also acknowledge the contribution of all the developers and users of the tools, as documented in the GitHub repositories.

References

[1] Siala K, Houmy H. tum-ens/pyGRETA: python generator of REnewable Time series and mAps. Zenodo; 2020, Version v1.1.0. http://dx.doi.org/10.5281/zenodo.3727416.

[2] Siala K, Houmy H, Khan WS, Mahfouz MY. tum-ens/pyCLARA: python Clustering of Lines And RAsters. Zenodo; 2020, version v1.0.1. http://dx.doi.org/10.5281/zenodo.3872273.

[3] Siala K, Houmy H. tum-ens/pyPRIMA: python preprocessing of inputs for model frAmeworks. Zenodo; 2020, Version v1.0.1. http://dx.doi.org/10.5281/zenodo.3872022.

[4] Pfenninger S, Hawkes A, Keirstead J. Energy systems modeling for twenty-first century energy challenges. Renew Sustain Energy Rev 2014;33:74–86. http://dx.doi.org/10.1016/j.rser.2014.02.003.

[5] Klein S, Klein SW, Steinert T, Fricke A, Peschel D. In: enervis energy advisors GmbH, editor. Erneuerbare Gase - ein Systemupdate der Energiewende [Renewable gases - a system update of the energy transition]. Tech. rep., INES Initiative Erdgasspeicher e.V. / BWE Bundesverband Windenergie e.V.; 2017, URL https://www.wind-energie.de/fileadmin/redaktion/dokumente/publikationen-oeffentlich/themen/03-sektorenkopplung/20171212_studie_erneuerbare_gase.pdf. Last viewed: May 31, 2020.

[6] Benz T, Dickert J, Erbert M, Erdmann N, Johae C, Katzenbach B, Glaunsinger W, Müller H, Schegner P, Schwarz J, Speh R, Stagge H, Zdrallek M. Der zelluläre Ansatz: Grundlage einer erfolgreichen, regionenübergreifenden Energiewende [The cellular approach: Basis for a successful, cross-regional energy transition]. Tech. rep., ETG-VDE; 2015, URL https://www.bund-naturschutz.de/fileadmin/Bilder_und_Dokumente/Presse_und_Aktuelles/Pressemitteilungen/2017/Energie_und_Klima/VDE_ST_ETG_ZellulareAnsatz_web.pdf. Last viewed: May 31, 2020..

[7] Amado M, Poggi F, Ribeiro Amado A, Breu S. A cellular approach to net-zero energy cities. Energies 2017;10(11):1826. http://dx.doi.org/10.3390/en10111826.

[8] Hörsch J, Brown T. The role of spatial scale in joint optimisations of generation and transmission for European highly renewable scenarios. In: 2017 14th international conference on the european energy market (EEM). IEEE; 2017, p. 1–7. http://dx.doi.org/10.1109/EEM.2017.7982024.

[9] Anderski T, Surmann Y, Stemmer S, Grisey N, Momot E, Leger A-C, Betraoui B, van Roy P. e-HIGHWAY 2050: D 2.2 - European cluster model of the Pan-European transmission grid. Tech. rep., European Union; 2015.

[10] Cao K-K, von Krbek K, Wetzel M, Cebulla F, Schreck S. Classification and evaluation of concepts for improving the performance of applied energy system optimization models. Energies 2019;12(24):4656. http://dx.doi.org/10.3390/en12244656.

[11] Frysztacki MM, Hörsch J, Hagenmeyer V, Brown T. The strong effect of network resolution on electricity system models with high shares of wind and solar. Appl Energy 2021;291:116726. http://dx.doi.org/10.1016/j.apenergy.2021.116726, URL https://www.sciencedirect.com/science/article/pii/S0306261921002439.

[12] Simoes S, Nijs W, Ruiz P, Sgobbi A, Thiel C. Decarbonised pathways for a low carbon EU28 power sector until 2050. In: 11th international conference on the European energy market (EEM14). 2014, p. 1–5. http://dx.doi.org/10.1109/EEM.2014.6861232.

[13] Kanellopoulos K, Kavvadias K, Careri F, De Felice M, Busch S, Hidalgo Gonzalez I. The operation of the European power system in 2016: A benchmarking study with METIS. EU Publ 2019. http://dx.doi.org/10.2760/434010, Reference number: KJ-NA-30008-EN-N.

[14] Mantzos L, Matei NA, Rózsai M, Russ P, Ramirez AS. POTEnCIA: A new EU-wide energy sector model. In: 2017 14th international conference on the European Energy Market (EEM). 2017, p. 1–5. http://dx.doi.org/10.1109/EEM.2017.7982028.

[15] Rey Los Santos L, Temursho U, Tamba M, Vandyck T, Wojtowicz K, Saveyn B, Weitzel M. Global macroeconomic balances for mid-century climate analyses. EU Publ 2018. http://dx.doi.org/10.2760/858469, Reference number: KJ-NA-29490-EN-N..

[16] Howells M, Rogner H, Strachan N, Heaps C, Huntington H, Kypreos S, Hughes A, Silveira S, DeCarolis J, Bazillian M, Roehrl A. OSeMOSYS: The open source energy modeling system. Energy Policy 2011;39(10):5850–70. http://dx.doi.org/10.1016/j.enpol.2011.06.033.

[17] Hörsch J, Hofmann F, Schlachtberger D, Brown T. PyPSA-Eur: An open optimisation model of the European transmission system. Energy Strategy Rev 2018;22:207–15. http://dx.doi.org/10.1016/j.esr.2018.08.012, URL https://www.sciencedirect.com/science/article/pii/S2211467X18300804.

[18] Hofmann F, Hampp J, Neumann F, Brown T, Hörsch J. atlite: A Lightweight python package for calculating renewable power potentials and time series. J Open Source Softw 2021;6(62):3294. http://dx.doi.org/10.21105/joss.03294.

[19] Ryberg D, Robinius M, Stolten D. Evaluating land eligibility constraints of renewable energy sources in europe. Energies 2018;11(5):1246. http://dx.doi.org/10.3390/en11051246.

[20] Ryberg D, Tulemat Z, Robinius M, Stolten D. Geospatial land availability for energy systems (GLAES). 2018, Available online: https://github.com/FZJ-IEK3-VSA/glaes/releases/tag/v1.2.0, last viewed: June 30, 2021.

[21] International Renewable Energy Agency. Global Atlas. Global Atlas Consortium; 2019, URL http://www.irena.org/GlobalAtlas. Last viewed: May 31, 2020.

[22] Technical University of Denmark. Global Wind Atlas. World Bank Group; 2019, URL https://globalwindatlas.info. Last viewed: May 31, 2020.

[23] Solargis. Global Solar Atlas. World Bank Group; 2019, URL https://globalsolaratlas.info. Last viewed: May 31, 2020.

[24] Rey SJ, Anselin L. PySAL: A Python library of spatial analytical methods. In: Fischer MM, Getis A, editors. Handbook of applied spatial analysis: software tools, methods and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, p. 175–93. http://dx.doi.org/10.1007/978-3-642-03647-7_11.

[25] Hirth L, Mühlenpfordt J, Schlecht I, Reimann J, Weibezahn J, Weinhold R, Kendziorski M, Pfenninger S, Schill W-P. Open power system data. 2019, URL https://github.com/Open-Power-System-Data. last viewed: July 09, 2021.

[26] Huppmann D, Gidden M, Nicholls Z, Hörsch J, Lamboll R, Kishimoto P, Burandt T, Fricko O, Byers E, Kikstra J, Brinkerink M, Budzinski M, Maczek F, Zwickl-Bernhard S, Welder L, Álvarez Quispe E, Smith C. Pyam: Analysis and visualisation of integrated assessment and macro-energy scenarios [version 1; peer review: 1 approved with reservations]. Open Res Eur 2021;1(74). http://dx.doi.org/10.12688/openreseurope.13633.1.

[27] Siala K, Mahfouz MY. Impact of the choice of regions on energy system models. Energy Strategy Rev 2019;25:75–85. http://dx.doi.org/10.1016/j.esr.2019.100362.

[28] Siala K, de la Rúa C, Lechón Y, Hamacher T. Towards a sustainable European energy system: Linking optimization models with multi-regional input-output analysis. Energy Strategy Rev 2019;26:100391. http://dx.doi.org/10.1016/j.esr.2019.100391.

[29] Siala K, Chowdhury AK, Dang TD, Galelli S. Solar energy and regional coordination as a feasible alternative to large hydropower in Southeast Asia. Nature Commun 2021;12(4159). http://dx.doi.org/10.1038/s41467-021-24437-6.

[30] Siala K, Odersky L. Code and data for publication "pyGRETA, pyCLARA, pyPRIMA: A pre-processing suite to generate flexible model regions for energy system models". Zenodo; 2021, http://dx.doi.org/10.5281/zenodo.5054420.