PyPSA: Free Software for Investment Planning in Sector-Coupled Energy Systems

Fabian Neumann and Tom Brown

fabian.neumann@kit.edu
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology (KIT), Germany

based on many contributions from
J. Hörsch, F. Hofmann, M. Frysztacki, E. Zeyen, V. Hagenmeyer, M. Victoria, K. Zhu, M. Greiner

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

IEEE PES GM 2020
July 15, 2020

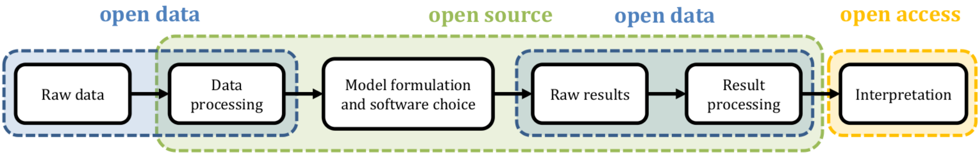
Karlsruhe Institute of Technology

Our Group's Research

- **Cost-effective pathways** to reduce greenhouse gas emissions
- **Co-optimisation** of generation, storage and transmission infrastructure (trade-offs)
- **Power grid reinforcement** requirements with large shares of wind and solar generation
- **Algorithms** to increase the tractability of energy system optimisation (spatial, temporal clustering, model reduction)
- **Open Source** software development and open data

We aim to follow the Idea(l) of Open Energy Modelling

The whole chain from raw data to modelling results should be open:



open data + free software ⇒ transparency + reproducibility

There's an initiative for that! Sign up for the mailing list / come to the next (virtual) workshop:

openmod open energy modelling initiative

http://openmod-initiative.org/

Source: openmod initiative

Agenda

1 Power System Analysis in Python with

PyPSA: Python for Power System Analysis

pypi v0.17.0 | conda-forge v0.17.0 | build passing | docs passing | license GPLv3 | DOI 10.5281/zenodo.786605 | chat on gitter

---

2 Data-Driven Modelling with

PyPSA-Eur: An Open Optimisation Model of the European Transmission System

release v0.2.0 | build passing | docs passing | repo size 25.9 MB | DOI 10.5281/zenodo.3520875 | chat on gitter

---

3 Sector-Coupling with

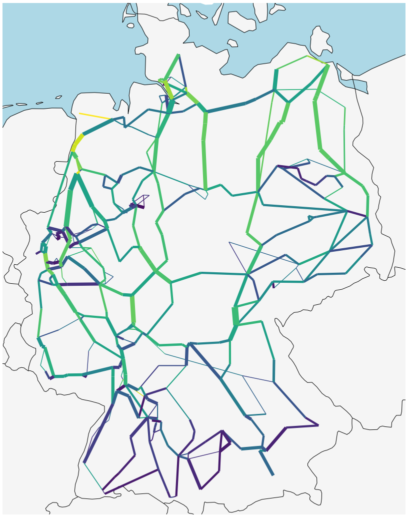
PyPSA-Eur-Sec: A Sector-Coupled Open Optimisation Model of the European Energy System

release v0.1.0 | docs passing | license GPL-3.0 | repo size 743 kB | chat on gitter

Karlsruhe Institute of Technology

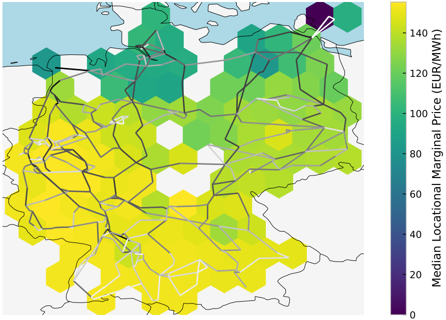
Python for Power System Analysis (PyPSA)

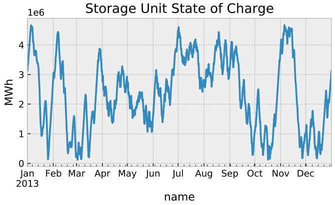
- **Open source** tool for modelling energy systems at **high resolution**.
- Fills missing gap between **load flow software** (e.g. PowerFactory, MATPOWER) and **energy system simulation software** (e.g. PLEXOS, TIMES, OSeMOSYS).
- Good grid modelling is increasingly important, for integration of **renewables** and **electrification** of transport, heating and industry.



PyPSA can do

- Static **power flow** (Newton-Raphson)
- **Linear optimal power flow** (LOPF)
  - multiple periods
  - unit commitment
  - storage units
  - coupling to other sectors
- **Security-constrained LOPF**
- Coordinated total energy system **capacity expansion optimisation**





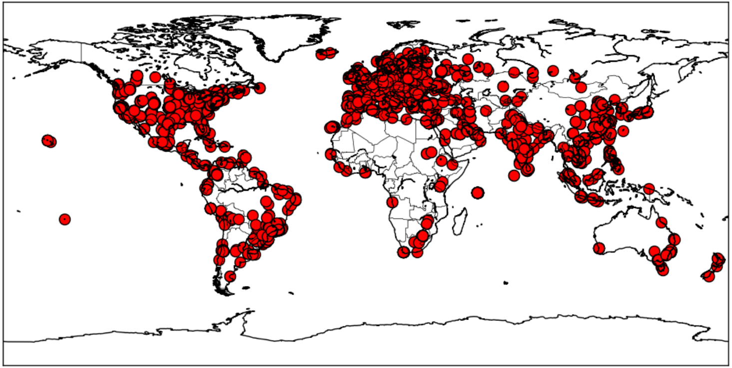
PyPSA has models for

- meshed **AC and DC networks**, with controllable converters
- **standard types** for lines and transformers following the implementation in pandapower
- conventional **dispatchable generators** with **unit commitment**
- generators with **time-varying power availability**, such as wind and solar generators
- **storage units** with efficiency losses; simple **hydroelectricity** with inflow and spillage
- coupling of **other energy carriers**, e.g. P2G, P2L, P2H, DAC, district heating, gas networks, electric vehicles, carbon capture and usage/sequestration (CCU/S)
- **easy to extend**: build complicated assets out of basic components

Worldwide Usage

PyPSA is used worldwide by **dozens of research institutes and companies**, e.g.

- TU Delft
- TransnetBW (TSO)
- Fraunhofer ISE
- Fraunhofer IEG
- DLR Oldenburg
- Shell
- Saudi Aramco
- Reiner Lemoine Institute
- HS Flensburg
- Aarhus University
- Jülich Research Centre
- Chalmers University of Technology
- Frankfurt Institute of Advanced Studies
- Karlsruhe Institute of Technology



Comparison to Other Software Tools

	Software	Language	Version	Free Software	Grid Analysis: Power Flow	Grid Analysis: Continuation Power Flow	Grid Analysis: Dynamic Analysis	Economic Analysis: Transport Model	Economic Analysis: Linear OPF	Economic Analysis: N-1 OPF	Economic Analysis: Nonlinear OPF	Economic Analysis: Multi-Period Optimisation	Economic Analysis: Unit Commitment	Economic Analysis: Investment Optimisation	Economic Analysis: Other Energy Sectors
Power system tools	NEPLAN	–	5.5.8		✓		✓	✓	✓	✓	✓				✓
Power system tools	PowerFactory	–	2017		✓		✓		✓	✓	✓				
Power system tools	PowerWorld	–	19		✓		✓	✓	✓	✓	✓				
Power system tools	PSS/E	–	33.10		✓		✓		✓	✓	✓				
Power system tools	PSAT	MATLAB	2.1.11	✓	✓	✓	✓		✓		✓	✓	✓		
Power system tools	MATPOWER	MATLAB	7.0	✓	✓	✓		✓	✓		✓				
Power system tools	PYPOWER	Python	5.1.4	✓	✓			✓	✓		✓				
Power system tools	pandapower	Python	2.2.2	✓	✓			✓	✓		✓				
Power system tools	PowerDynamics.jl	Julia	2.3.1	✓			✓								
Power system tools	PowerSimulations.jl	Julia	0.5.5	✓				✓	✓		✓	✓	✓		
Power system tools	PowerModels.jl	Julia	0.17.2	✓	✓				✓		✓	(✓)	✓		
	PyPSA	Python	0.17.0	✓	✓			✓	✓	✓		✓	✓	✓	✓
Energy system tools	calliope	Python	0.5.2	✓				✓				✓		✓	✓
Energy system tools	oemof	Python	0.4.0	✓				✓				✓	✓	✓	✓
Energy system tools	OSeMOSYS	various	2020	✓				✓				✓		✓	✓
Energy system tools	PowerGAMA	Python	1.1	✓				✓	✓			✓			
Energy system tools	TIMES	GAMS	4.4.2	(✓)				✓	✓			✓	✓	✓	✓
Energy system tools	PLEXOS	–	8.1					✓	✓	✓		✓	✓	✓	✓
Energy system tools	PRIMES	–	2017					✓	✓			✓	✓	✓	✓

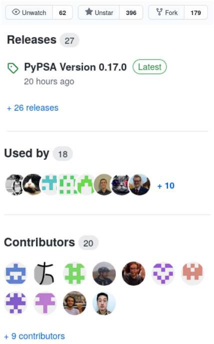
Technical Details

PyPSA leans heavily on

- **pandas** for storing data about components and time series
- **numpy** and **scipy** for linear algebra and sparse matrices
- **networkx** for some network calculations
- **pyomo** for preparing optimisation problems
- **cartopy** for plotting networks on a map
- **py.test** for unit testing (PYPOWER and pandapower)

Recent and current developments include

- **custom optimisation framework** as alternative to pyomo
- **multi-horizon** investment optimisation



Simple Example

```
import pypsa

n = pypsa.Network("mynetwork.nc")

n.lopf(solver_name="cbc", pyomo=False, solver_options={"primalT": 1e-4})

n.generators.p_nom_opt                      # static attributes

n.generators_t.p                            # time-dependent attributes

n.plot(geomap=True, line_colors=n.lines_t.p0.abs().mean())

n.pf()                                      # AC power flow
```

Example Problem: Greenfield Power System Investment Planning

Find the long-term cost-optimal energy system, including investments and short-term costs:

$$\textbf{Min}\left[\begin{array}{c}\textbf{Yearly}\\\textbf{system costs}\end{array}\right] = \textbf{Min}\left[\sum_{n}\left(\begin{array}{c}\textbf{Annualised}\\\textbf{capital costs}\end{array}\right) + \sum_{n,t}\left(\begin{array}{c}\textbf{Marginal}\\\textbf{costs}\end{array}\right)\right]$$

subject to

- meeting energy demand at each node n (e.g. region) and time t (e.g. hour of year)
- transmission constraints between nodes and linearised power flow
- wind, solar, hydro (variable renewables) availability time series $\forall\ n, t$
- installed capacity $\leq$ geographical potentials for renewables
- fulfilling CO_2 emission reduction targets

Challenges of Data-Driven Modelling

- Many different data **sources**
- Many data sources need **cleaning** and processing before they can be used
- Many **intermediate scripts** and datasets
- Often **dependencies** are not clear (both data and software)
- Data and code **change** over time
- Want to be able to **reproduce** results
- Want to run many **parametric scenarios** for same model

Workflow Management Tool `snakemake`

Snakefile

```
rule mytask:
    input:
        "data/{sample}.txt"
    output:
        "result/{sample}.txt"
    script:
        "scripts/mytask.py"
```

```
rule myplot:
    input:
        "result/{sample}.txt"
    output:
        "figures/{sample}.pdf"
    script:
        "scripts/myplot.py"
```

command: `$ snakemake figures/greatstuff.pdf`

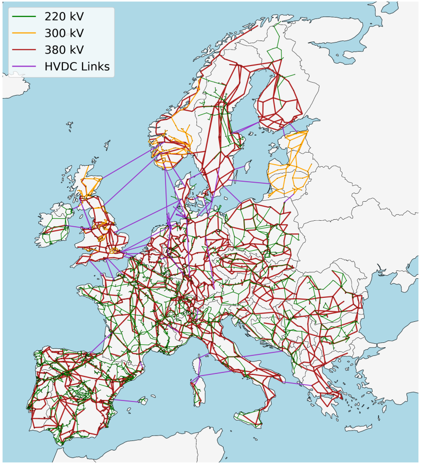
Support for Bash, Python, Julia, and R scripts. Neat cluster integration.

What is PyPSA-Eur?

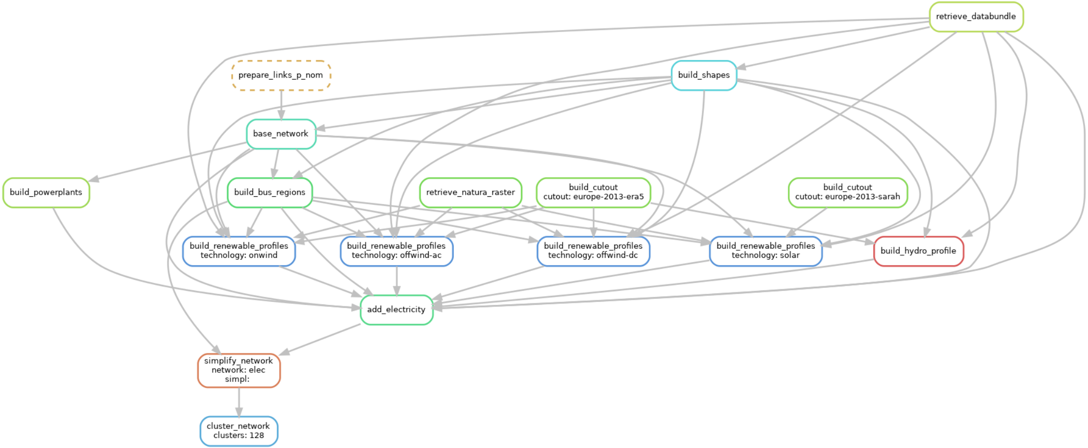
An automated and configurable workflow to build a model of the European power system from open data.

Covers the ENTSO-E area and contains

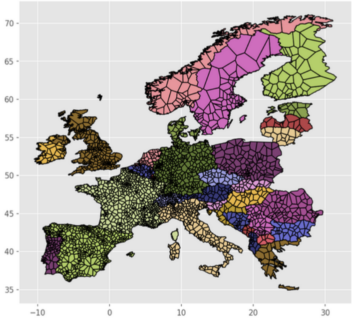
- all AC lines at and above 220 kV, substations and HVDC links,
- a database of conventional power plants,
- time series for electrical demand,
- time series for variable renewable generator availability, and
- geographic potentials for the expansion of wind and solar power.



Snakemake for PyPSA-Eur: Building Model



Installable Potential and Land Eligibility



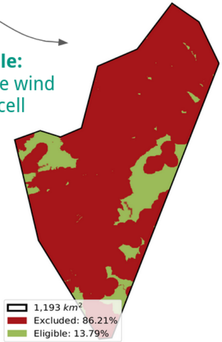
example:
onshore wind
in one cell

Geospatial Land Availability for Energy Systems (GLAES)

JÜLICH FORSCHUNGSZENTRUM

DOI 10.5281/zenodo.1122558

Severin Ryberg
github.com/FZJ-IEK3-VSA/glaes



- **CORINE 2018** land cover
  - eligible codes
  - distances
- **NATURA 2000** natural protection areas
- **GEBCO 2018** bathymetry dataset
- **Density:** capacity per km²

 17

KIT Karlsruhe Institute of Technology

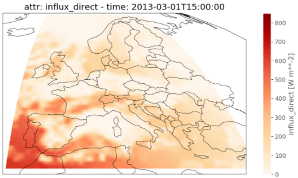
Derivation of Time Series of Variable Renewables

Reanalysis weather data
from **ERA-5** and **SARAH-2**
(up to 40 years, 30x30 km)

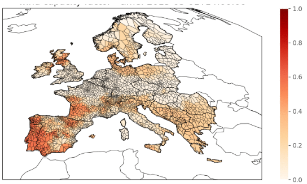
https://atlite.readthedocs.io/
https://github.com/PyPSA/atlite

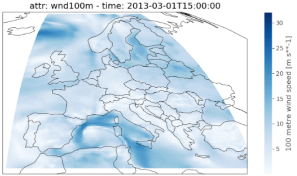
Atlite: Convert weather data to energy systems data

pypi v0.0.2 | conda-forge v0.0.2 | docs passing | license GPLv3

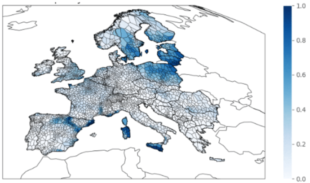


Solar panel models
- orientation
- material



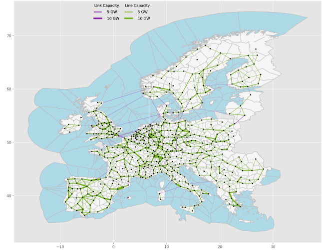


Wind turbine models
- power curve
- surface roughness

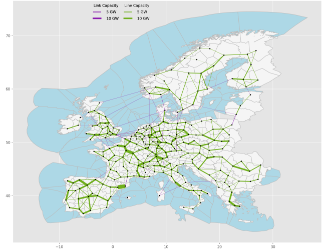


Karlsruhe Institute of Technology

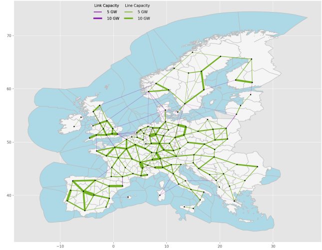
Network Clustering



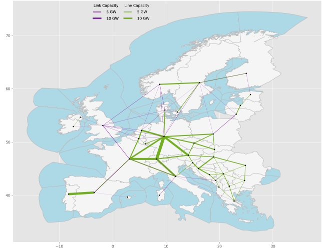
512 nodes



256 nodes



128 nodes



37 nodes

Snakemake for PyPSA-Eur: Managing Scenarios

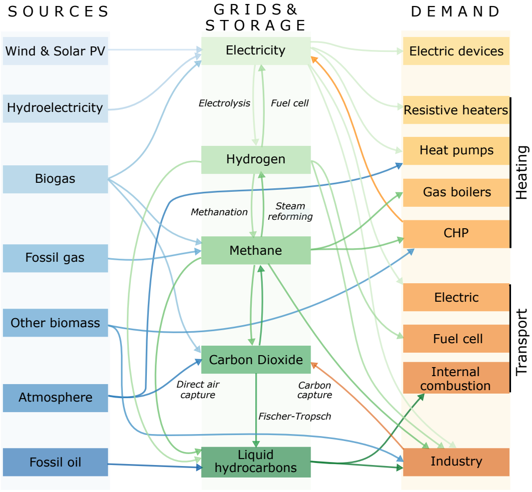
What is PyPSA-Eur-Sec?

Need to decarbonize all sectors in Europe obeying spatial and temporal constraints.

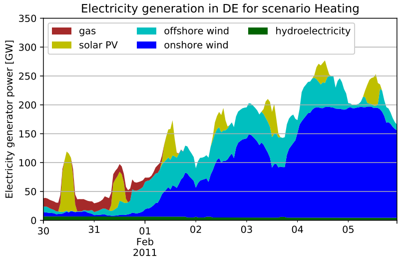
Extends **PyPSA-Eur** power system model with

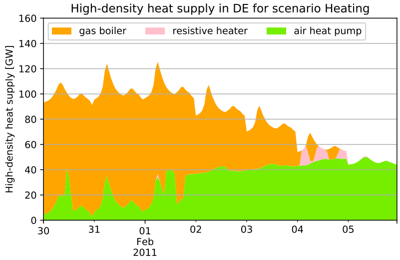
- transport sector
- heating sector
- industry sector
- aviation
- shipping
- industrial feedstocks
- biomass
- better carbon management
- carbon capture and usage/sequestration
- gas networks

⟶ multiple inputs/outputs



Example problem with balancing: Cold week in winter





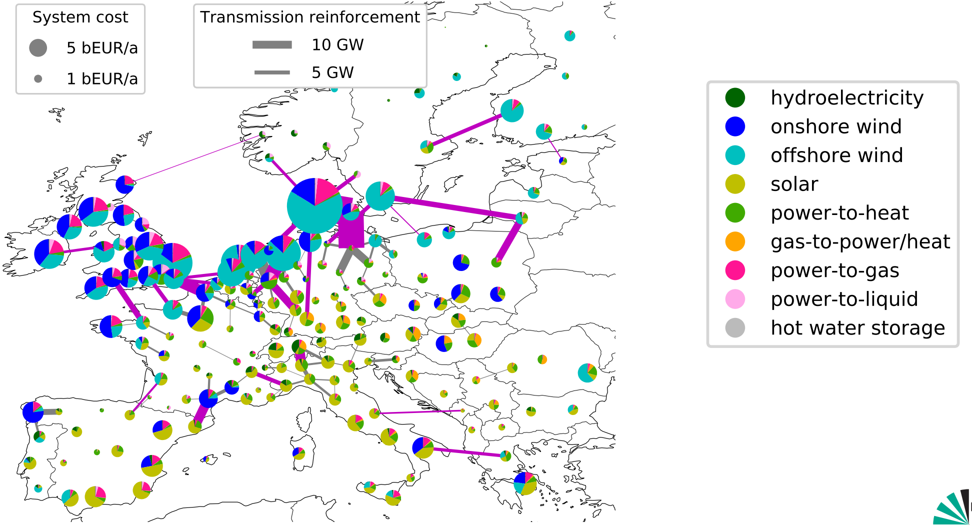
There are difficult periods in winter with:

- **Low** wind and solar generation
- **High** space heating demand
- **Low** air temperatures, which are bad for air-sourced heat pump performance

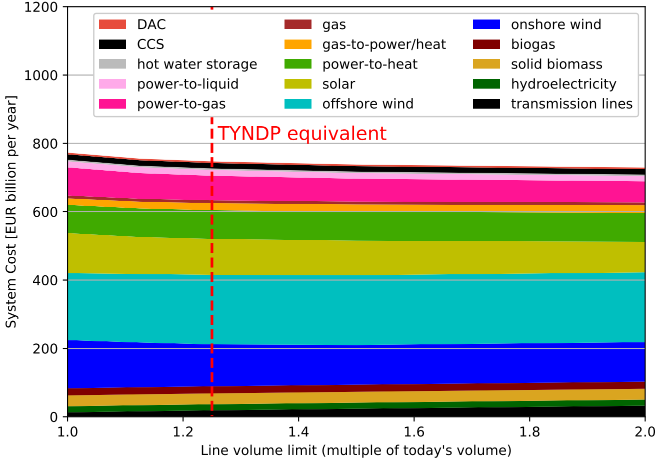
Less-smart solution: **backup gas boilers** burning either natural gas, or synthetic methane.

Smart solution: **long-term thermal energy storage** in **district heating networks** and efficient **combined-heat-and-power plants**.

Preliminary: Zero Emissions – 25% more grid volume (TYNDP)



Preliminary: Benefit of grid expansion for sector-coupled system



- Direct system costs **bit higher than today's system**
- Systems **without grid expansion** are feasible, but more costly
- As grid is expanded, **costs reduce** from solar and P2G; more offshore wind
- Total cost benefit of extra grid: $\sim$ € 47 billion per year
- **Over half of benefit available at 25% expansion** (like TYNDP)
- easy to implement with `snakemake`

Conclusion

PyPSA is for researchers, planners and utilities who need an easy-to-use tool for power system analysis.

Not so useful for

- dynamic analysis (differential equations)
- short-circuit calculations
- multiphase and unbalanced load flow
- low-voltage distribution grids

Useful for

- investment planning with high spatial and temporal scope and detail
- electricity market analysis
- coupling to other sectors
- adding custom constraints and components

PyPSA-Eur and PyPSA-Eur-Sec

Configurable workflows to build detailed power/energy system models from open data that can be used for both operational and investment planning studies.

Karlsruhe Institute of Technology

Resources

Find the slides:

https://neumann.fyi/files/pesgm2020.pdf

Reach out:

fabian.neumann@kit.edu

Find the energy system model:

Code: https://github.com/pypsa/pypsa-eur

Documentation: https://pypsa-eur.readthedocs.io

Karlsruhe Institute of Technology