AMERICAN MADE SOLAR PRIZE

U.S. DEPARTMENT OF ENERGY

Virtual Tutorial Series

Open-Source Tools & Open-Access Solar Data

Webinar series part 3: Modeling Tools

Kevin Anderson, *Sandia National Laboratories*
Tassos Golnas, *Solar Energy Technologies Office (SETO), DOE*
Matt Prilliman, *National Renewable Energy Laboratory (NREL)*

Data: a Means to an End

Better photovoltaic (PV) models and system performance through high-quality data.

PV models are important in:

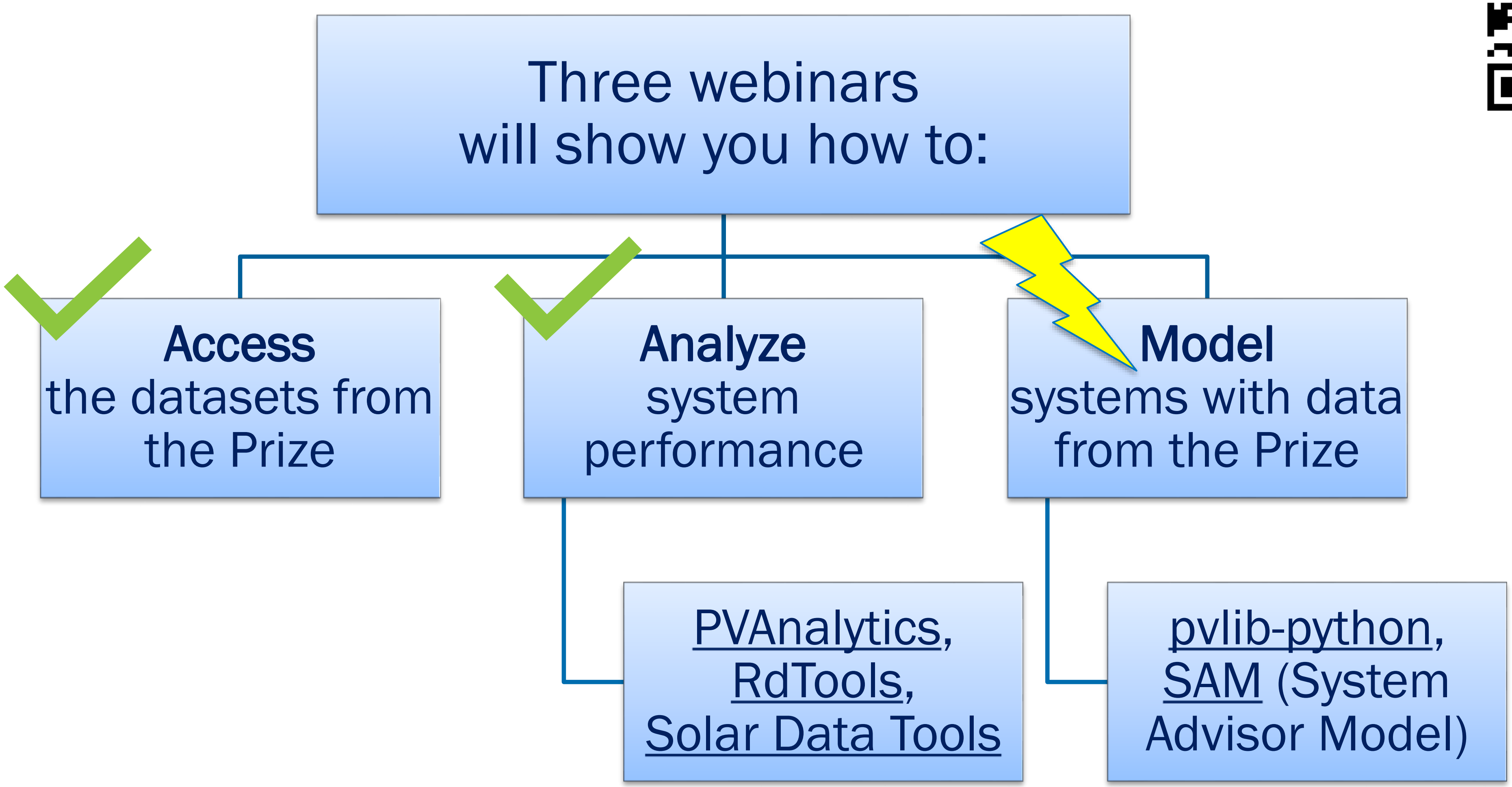
- Project development and valuation
- Power plant operation and maintenance

Better system performance means
lower cost of solar electricity

Prize goal:
Support industry and academic research efforts to **develop**, **improve**, **evaluate**, and **validate** models of real-world PV system performance in diverse locations.

Register!

Open-Access Data & Open-Source Tools



Three webinars will show you how to:

✓ **Access** the datasets from the Prize

✓ **Analyze** system performance

- PVAnalytics, RdTools, Solar Data Tools

Model systems with data from the Prize

- pvlib-python, SAM (System Advisor Model)

Modeling tutorial overview

We will be showing demos for two open-source PV modeling tools:

PySAM

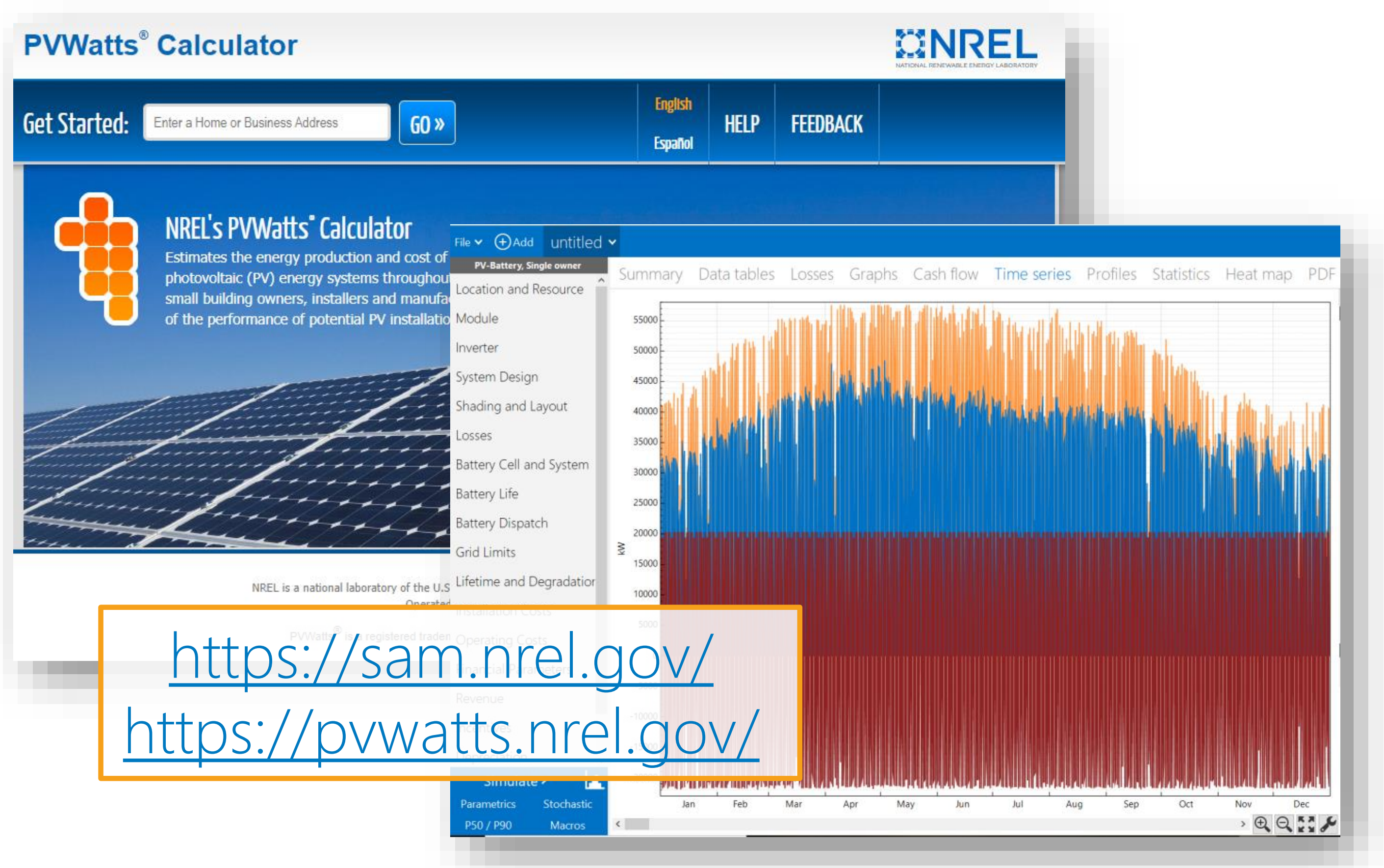
- Python interface to SAM
- Resource-to-energy simulations
- Can model many technologies (not just PV), plus financials

pvlib

- Flexible toolbox approach
- Provides several alternatives for each model type
- Useful in a variety of PV applications

System Advisor Model (SAM) & PVWatts

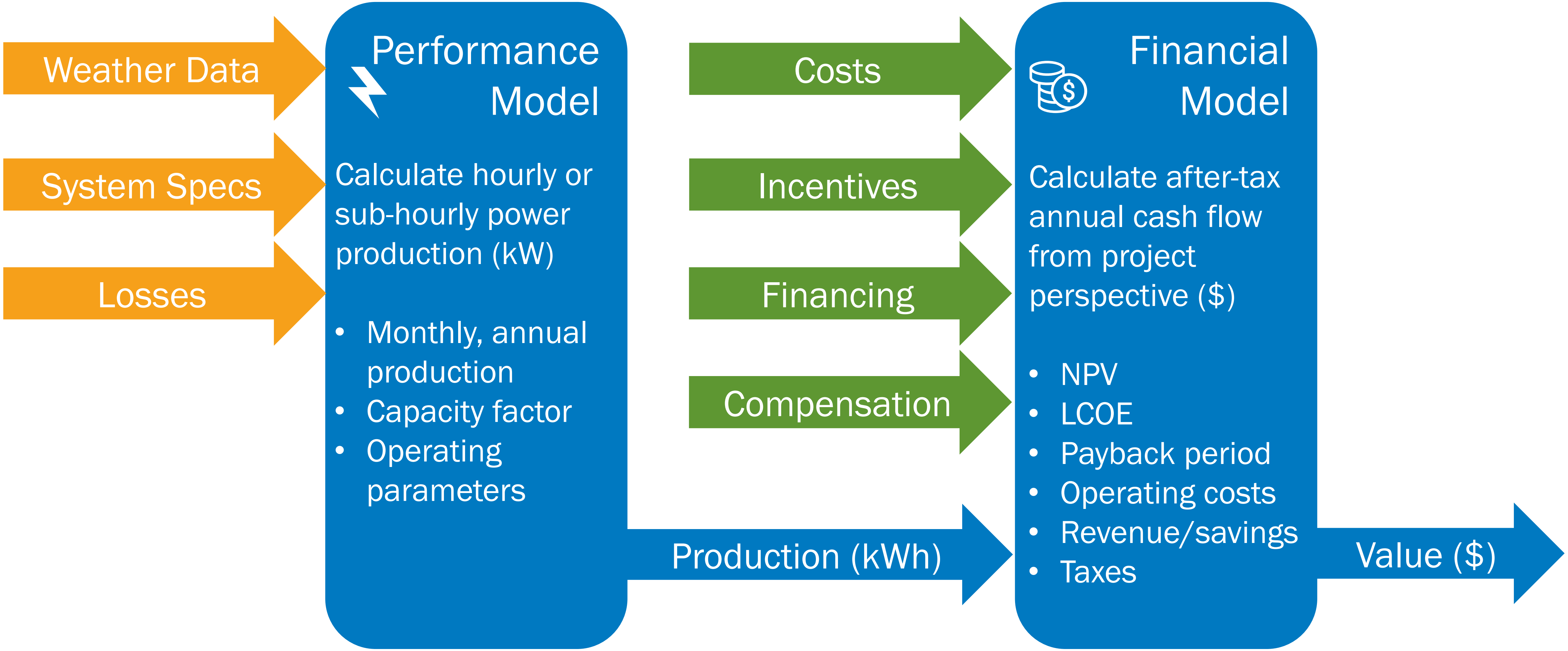
Free software that enable detailed performance and financial analysis for renewable energy systems



https://sam.nrel.gov/
https://pvwatts.nrel.gov/

- ✓ Desktop application
- ✓ PVWatts web tool & API
- ✓ Software development kit
- ✓ PySAM Python package
- ✓ Open source code
- ✓ Extensive documentation
- ✓ User support

Model Structure



Weather Data

System Specs

Losses

Performance Model

Calculate hourly or sub-hourly power production (kW)

- Monthly, annual production
- Capacity factor
- Operating parameters

Costs

Incentives

Financing

Compensation

Production (kWh)

Financial Model

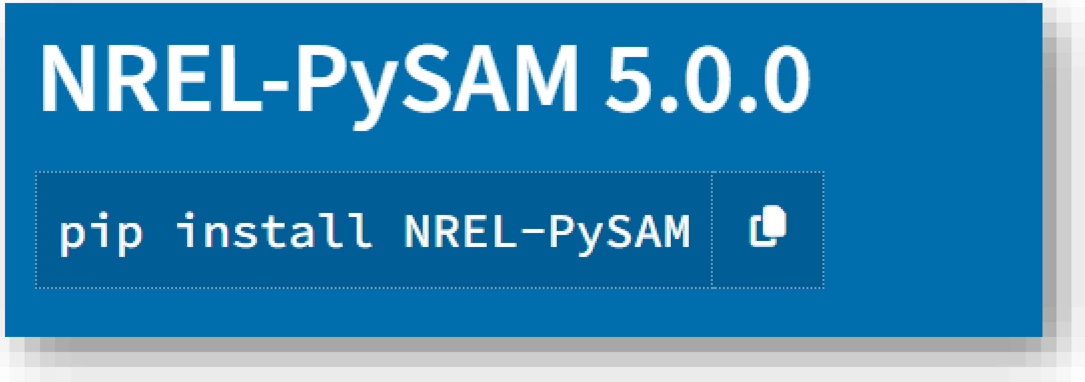
Calculate after-tax annual cash flow from project perspective ($)

- NPV
- LCOE
- Payback period
- Operating costs
- Revenue/savings
- Taxes

Value ($)

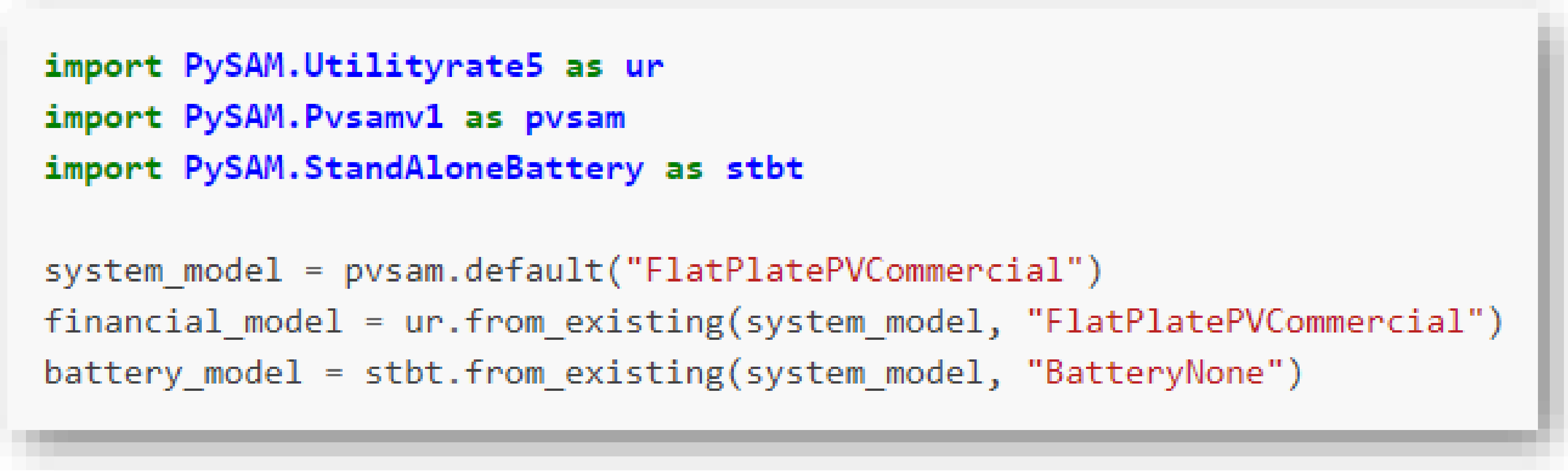
PySAM

- Python wrapper of SAM code
- Automatic code generation through SDK
- PyPi
- Documentation
- Github Repo



NREL-PySAM 5.0.0

```
pip install NREL-PySAM
```



```python
import PySAM.Utilityrate5 as ur
import PySAM.Pvsamv1 as pvsam
import PySAM.StandAloneBattery as stbt

system_model = pvsam.default("FlatPlatePVCommercial")
financial_model = ur.from_existing(system_model, "FlatPlatePVCommercial")
battery_model = stbt.from_existing(system_model, "BatteryNone")
```

What is pvlib?

A python library for PV performance modeling that is **community-driven**, **free**, **open-source**, and **well-documented**

REFERENCE MODELS

Stand-alone models for each step of the modeling chain

Transparent, peer-reviewed implementations

MODEL WORKFLOW

Weather-to-power following the PVPMC workflow

Customizable end-to-end PV system modeling (ModelChain)

DATA I/O

Parsing of standard file formats, e.g., TMY2, TMY3, EPW

Automated fetching of 12+ weather data sources

Selection of model implementations

pvlib.solarposition
- SPA
- ephemeris
- hour_angle

pvlib.irradiance
- Transposition models
- Decomposition models

pvlib.clearsky
- Ineichen
- Simplified solis

pvlib.snow
- Marion model
- Townsend

pvlib.bifacial
- infinite_sheds

pvlib.iam
- martin_ruiz
- martin_ruiz_diffuse
- marion_diffuse
- physical

pvlib.temperature
- faiman
- fuentes
- ross
- noct_sam
- pvsyst
- prilliman transient model

pvlib.ivtools
- fit_sde_sandia
- fit_pvsyst_sandia
- fit_desoto_sandia

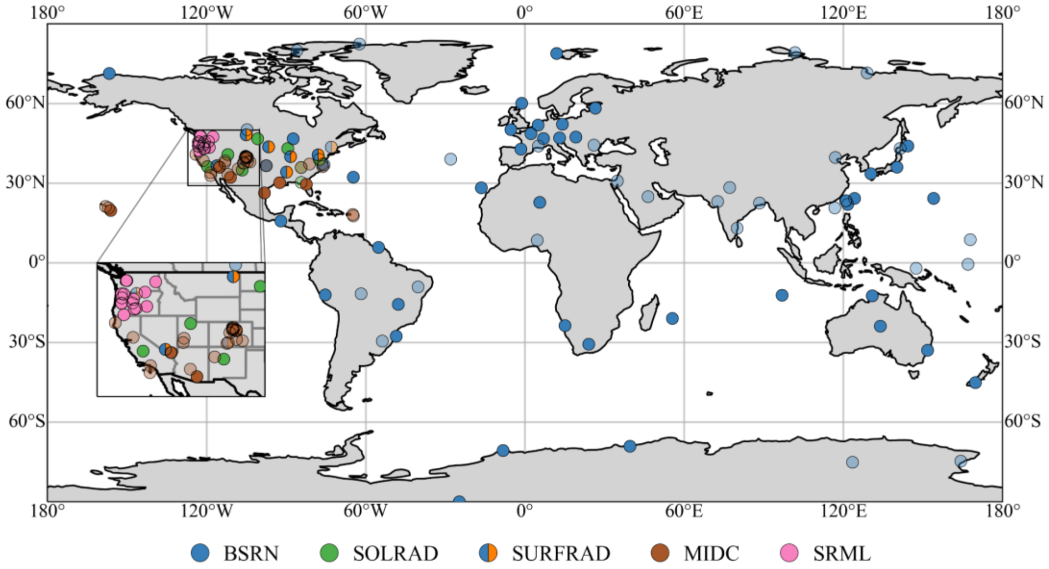
pvlib.soiling
- Kimber model
- Humboldt State model

pvlib.tracking
- slope-aware backtracking

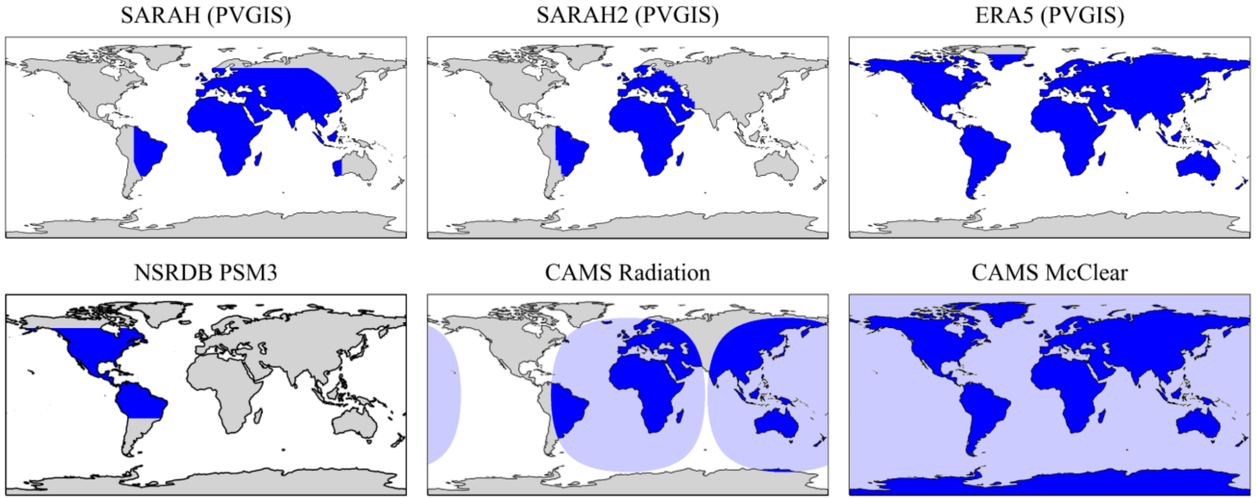
pvlib.inverter
- fit_sandia
- sandia_multi
- pvwatts_multi

pvlib.iotools: fetching weather data

Irradiance measurement stations



Satellite/reanalysis datasets



Where to find pvlib python

Installation:

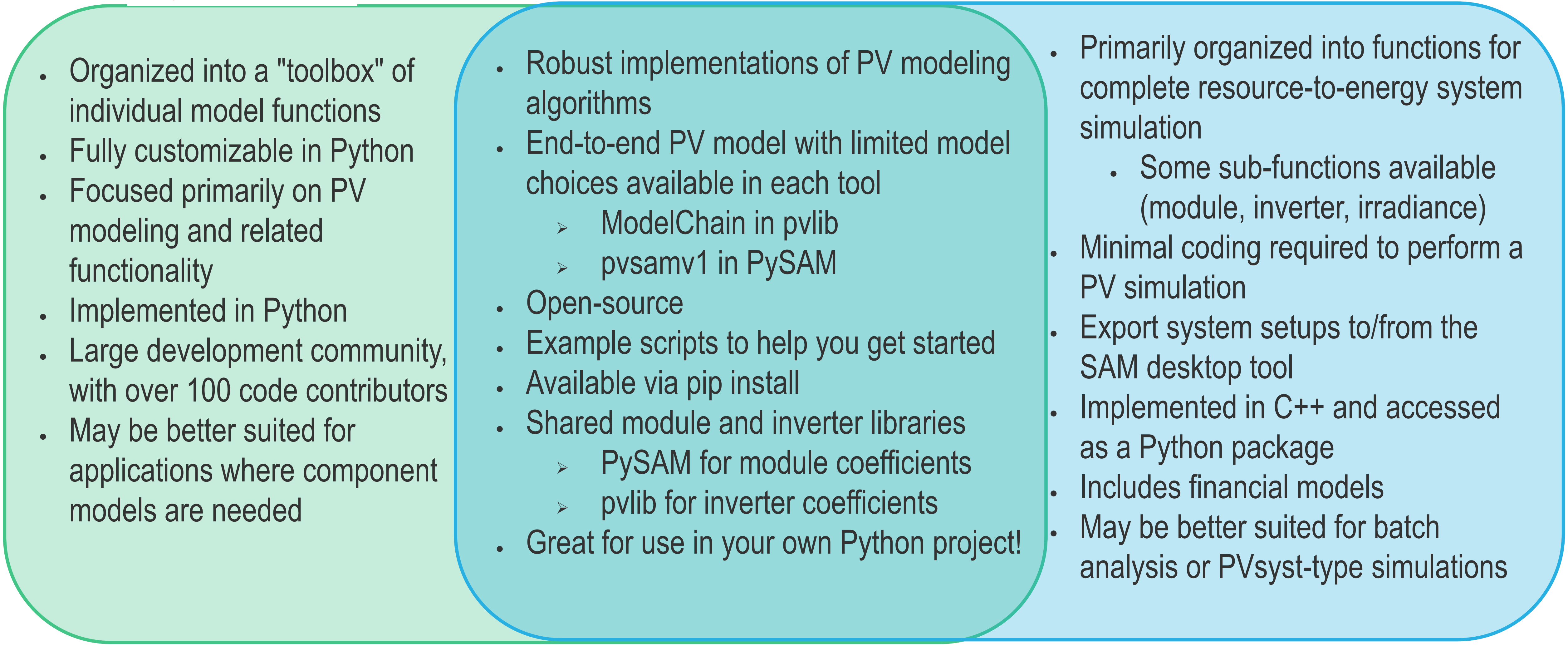
- Python Package Index: https://pypi.org/project/pvlib
  - pip install pvlib
- Conda-forge: https://anaconda.org/conda-forge/pvlib/
  - conda install –c conda-forge pvlib

Documentation: https://pvlib.readthedocs.io

Development: https://github.com/pvlib/pvlib-python

Google group: https://groups.google.com/g/pvlib-python

Demos



pvlib

- Organized into a "toolbox" of individual model functions
- Fully customizable in Python
- Focused primarily on PV modeling and related functionality
- Implemented in Python
- Large development community, with over 100 code contributors
- May be better suited for applications where component models are needed

Both!

- Robust implementations of PV modeling algorithms
- End-to-end PV model with limited model choices available in each tool
  - ➢ ModelChain in pvlib
  - ➢ pvsamv1 in PySAM
- Open-source
- Example scripts to help you get started
- Available via pip install
- Shared module and inverter libraries
  - ➢ PySAM for module coefficients
  - ➢ pvlib for inverter coefficients
- Great for use in your own Python project!

PySAM

- Primarily organized into functions for complete resource-to-energy system simulation
  - Some sub-functions available (module, inverter, irradiance)
- Minimal coding required to perform a PV simulation
- Export system setups to/from the SAM desktop tool
- Implemented in C++ and accessed as a Python package
- Includes financial models
- May be better suited for batch analysis or PVsyst-type simulations