Review of open source tools for PV modeling

Will Holmgren, U. Arizona

Clifford Hansen, Sandia National Lab

Joshua Stein, Sandia National Lab

Mark Mikofski, DNV-GL

Goals for this talk

- Promote the use of open source software in the PV modeling community
- Summarize the state of open source for PV modeling
- Stimulate discussion of how to support open source PV modeling projects in the future

Why use open source PV tools?

- *The Scientific Paper is Obsolete* (The Atlantic, April 2018)
- Encourages reproducibility and replicability in science
- Open source analyses encourage transparency and collaboration
- Project financing soft costs could be reduced through transparent, vetted algorithms
- More people looking at code, using it in different situations may yield more robust tools
- Modify the code to make it work better for you

https://github.com/wholmgren/openpvtools

Name	Purpose	Years Developed	Documentation Website	Development Website	Primary Languages	License
PVLib Matlab	General purpose PV modeling	2012 - *	pvpmc.sandia.gov	github.com/sandialabs/MATLAB_PV_LIB	Matlab	BSD 3
PVLib Python	General purpose PV modeling	2013 - *	pvlib-python.readthedocs.io	github.com/pvlib/pvlib-python	Python	BSD 3
System Advisory Model	Desktop app for PV, wind, CSP modeling, financial	2013 - *	sam.nrel.gov	github.com/NREL/SAM	C++	Mixed MIT/GPL 3
ssc	Compute modules for SAM	2010 - *	sam.nrel.gov	github.com/nrel/ssc	C, C++	Mixed MIT/GPL 3
rdtools	PV degradation	2017 - *	github.com/NREL/rdtools	github.com/NREL/rdtools	Python	MIT
PVFree	API for obtaining PV modeling parameters	2015 - *	pvfree.herokuapp.com	github.com/SunPower/pvfree	Python	Unlicensed
SolarUtils	Python wrappers of C solar position and spectral decomposition	2016	github.com/SunPower/SolarUtils	github.com/SunPower/SolarUtils	Python	BSD 3
Pecos	Performance monitoring	2016 - *	pecos.readthedocs.io	github.com/sandialabs/pecos	Python	BSD 3
Solpy	General purpose PV modeling	2011-2015	solpy.readthedocs.io	github.com/nrcharles/solpy	Python	LGPL 2.1
PVMismatch	IV curve calculator for mismatched cells	2012 - *	sunpower.github.io/PVMismatch/	github.com/SunPower/PVMismatch	Python	BSD 3
photovoltaic	General purpose PV modeling	2017 - *	github.com/trautsned/photovoltaic	github.com/trautsned/photovoltaic	Python	GPL 3
feedinlib	PV timeseries modeling	2015 - *	github.com/oemof/feedinlib	github.com/oemof/feedinlib	Python	GPL 3
CASSYS	PV system modeling	2015 - *	github.com/CanadianSolar/CASSYS	github.com/CanadianSolar/CASSYS	Excel, C#	BSD 3

Two development models

I giveth thee mostly-finished software that I've been toiling on in private

- SAM/ssc
- PVLIB Matlab
- Rdtools

Let's make some software in the open, warts and all

- PVLib Python
- PVMismatch
- feedinlib

My recommendation: choose what works for you, but don't be afraid to develop openly

Licenses

- Permissive: BSD 3, MIT
- Restrictive (copy left): GPL 3, LGPL 2.1
- Dual: Mixed GPL 3/MIT
- Unlicensed

- I urge you to:
  - Spend **15+ minutes** reading about licenses (multiple times)
  - Choose the license consistent with what **you** want – not your lawyers
  - License **all** of your code (unlicensed != public domain)

Funding

Funding

Source of funds

Kind of funds

	Public	**Private**
Direct Supports a specific open source tool	DOE support PVLib Matlab & SAM	Southern/EPRI funded UA to add solar forecasts to PVLib Python
Indirect Money/time spent on open source in pursuit of something else	DOE supports this workshop, which helps all of us.	SunPower, First Solar, DNV-GL, IMS, Sunshine Analytics... engineers contribute to open source software

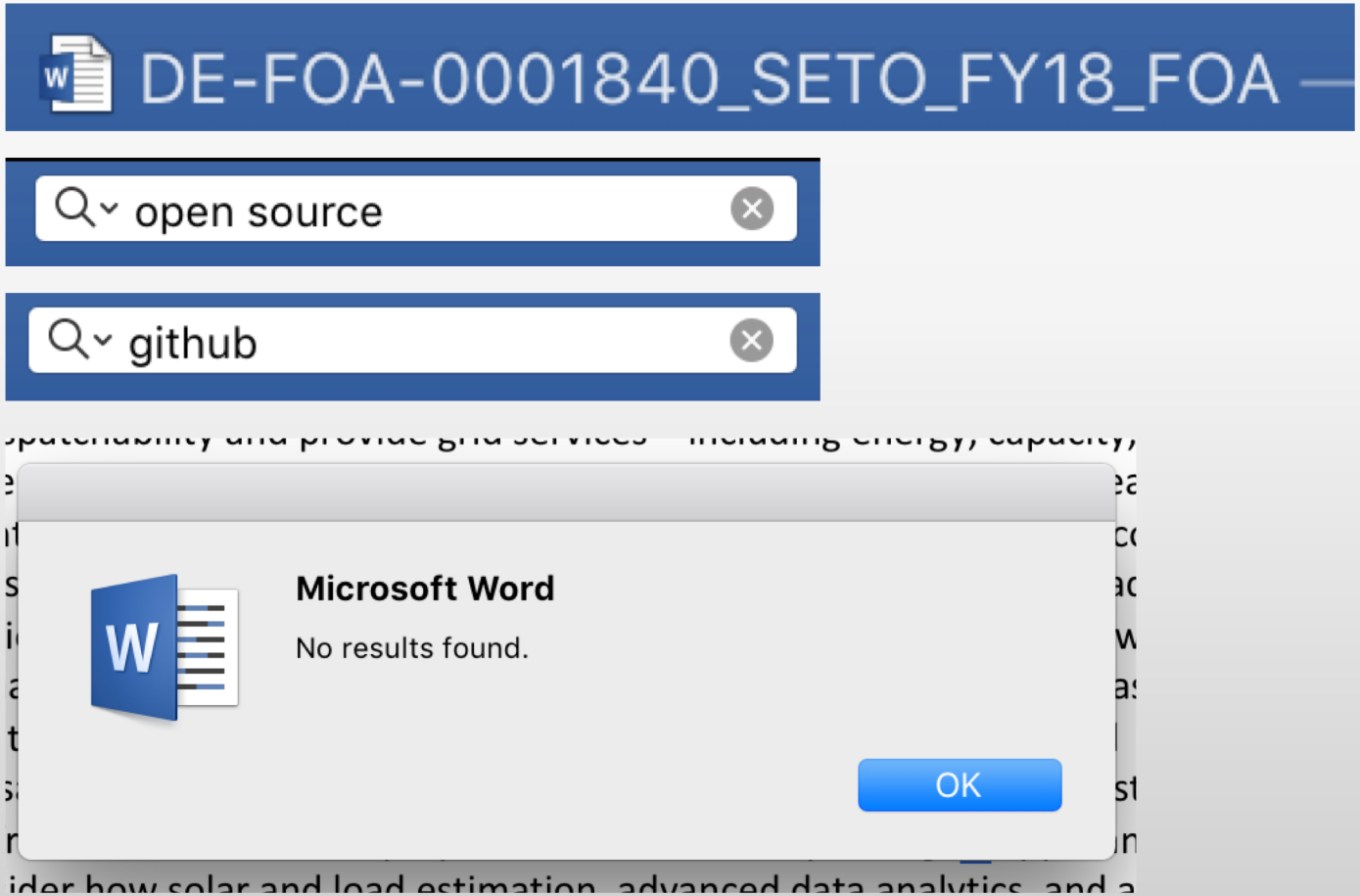
There are many ways you can support open source software!

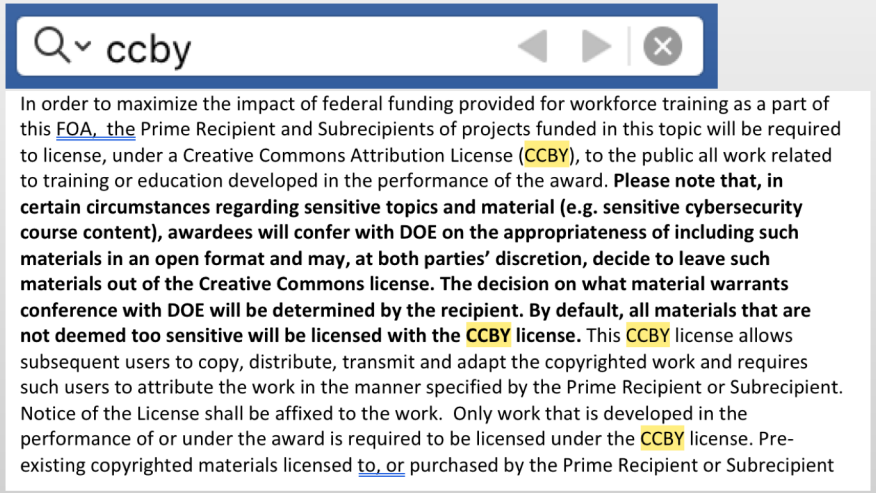
Funding

- *Show me the impact!* – person/group with the money
- Hard to trace impact
- Harder still to quantify impact

Funding

- Easy way for SETO to support open source:
  - FOA can require that software be released as open source
  - At least encourage it in the data management plan!

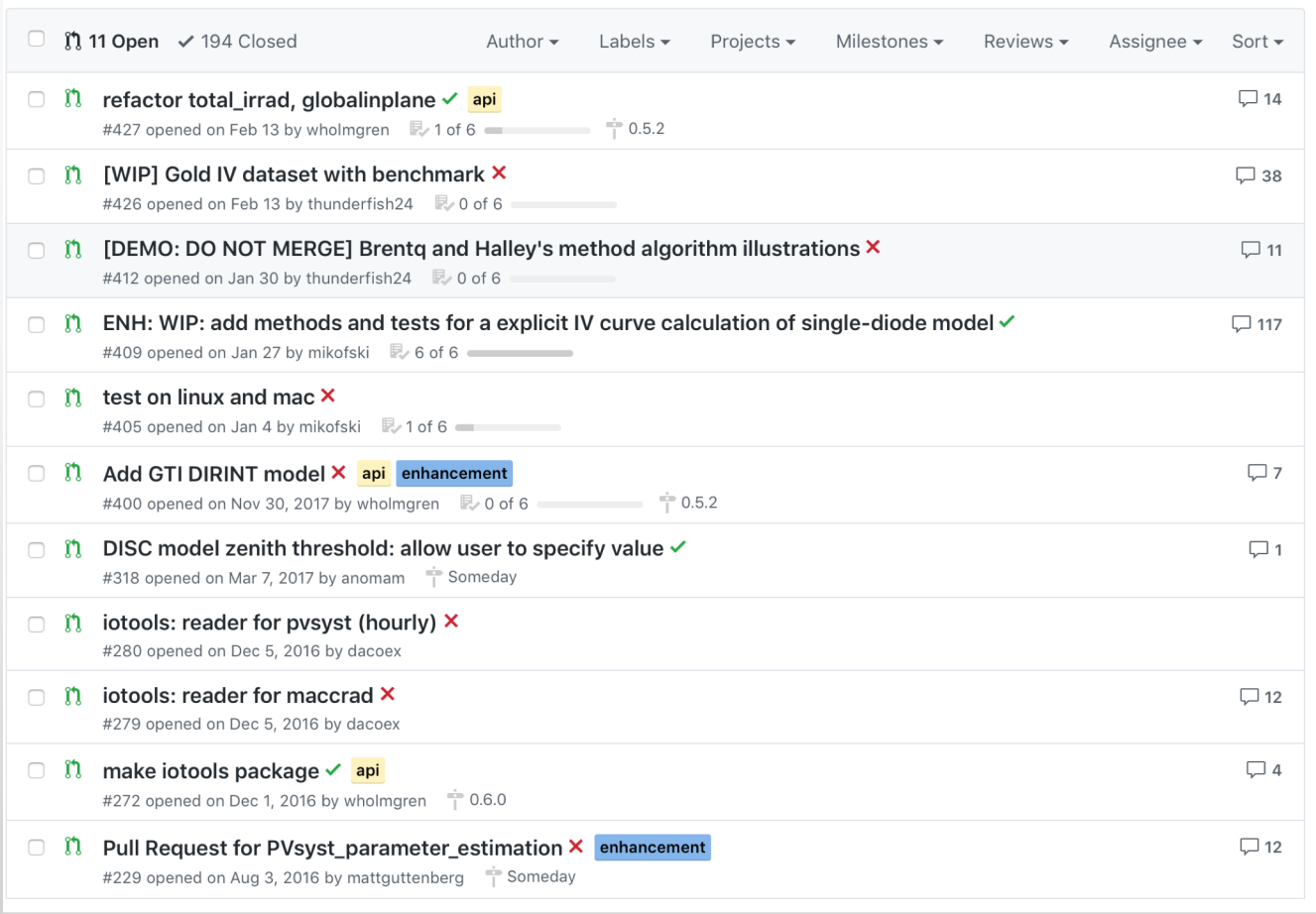




Community

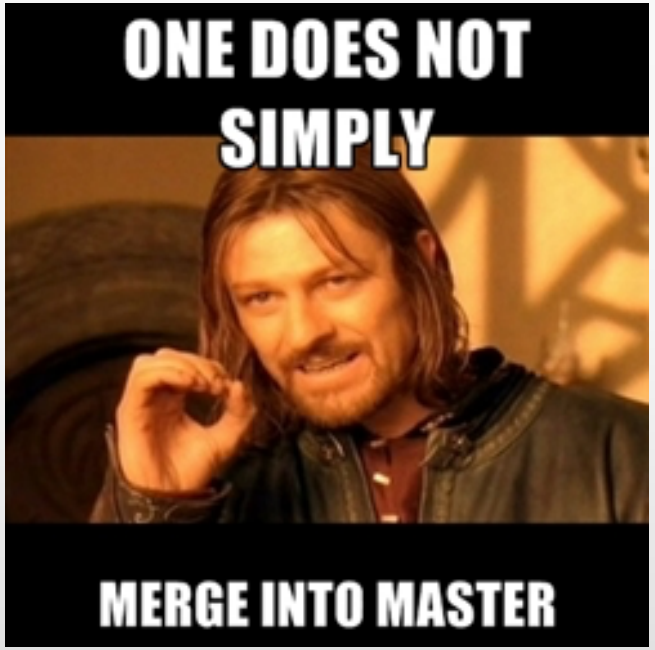
- Strong open source projects have strong user and developer communities
- Communities need help to grow and remain healthy
  - The PVPMC workshop is great for that
  - What else can we do?
- Who decides when code is ready to be merged into a package?
- Most people behave professionally, but not all
  - Formal codes of conduct may help

pvlib python open pull requests



Resources

- Version control, GitHub, package management stymies people – we will help you!
- But my code is no good – we will help you!
- Let's learn from others:
  - SciPy Conference
  - AMS Python Symposium
  - opensource.org
  - opensource.guide
  - *Roads and Bridges*, N. Eghbal
  - contributor-covenant.org



https://stackoverflow.com/questions/15651576/github-team-usage

Conclusions

- It’s wonderful that we can now have an open source PV review talk
- We should talk more about project scope, ambition, and collaboration, but...
- It’s also ok for open source projects to compete a little bit
- “Funding” for open source PV tools is complicated and evolving
- Future success or failure is determined by everyone in this room
  - Contribute as you can
  - Be respectful above all else

See Holmgren et. al., PVSC 2018 for more