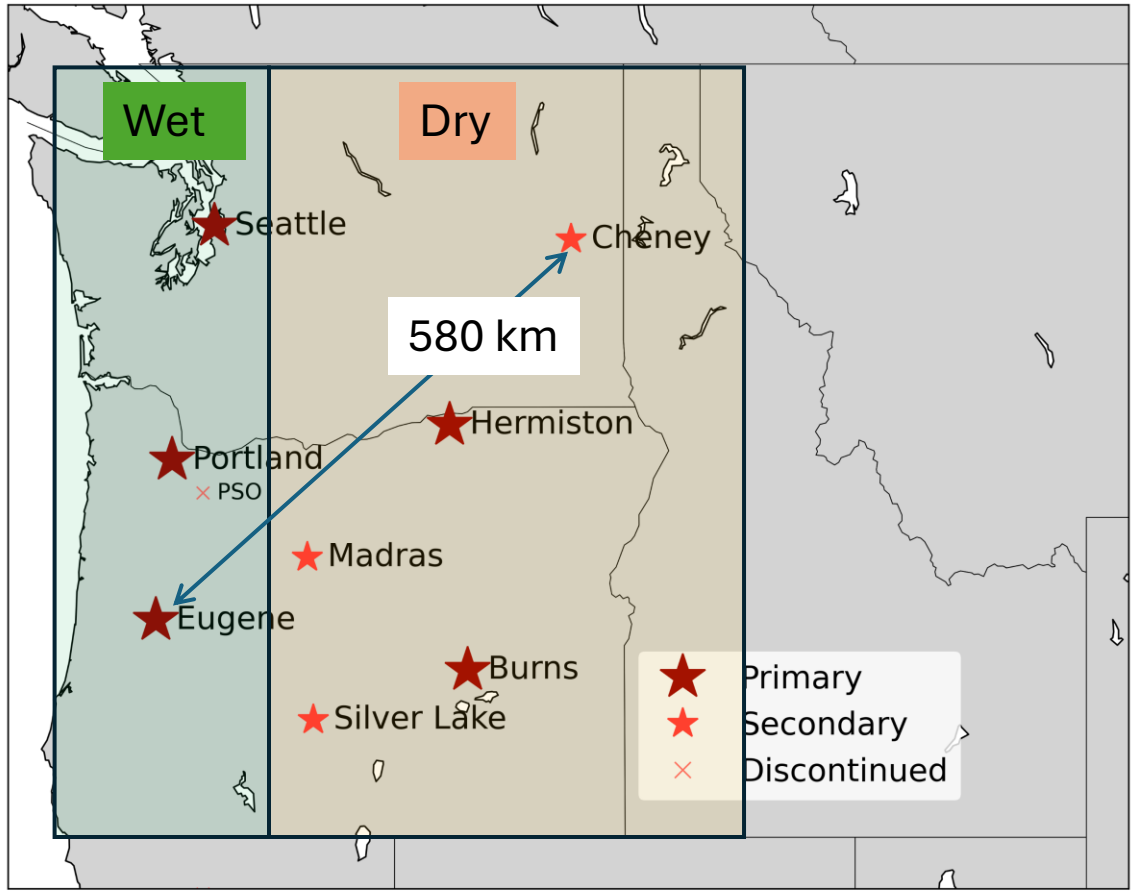
Operations of the UO Solar Radiation Network

Step by step overview the SRML data processing

Josh Peterson
University of Oregon
IPC, Davos Switzerland
2025-09-22 – 10-10

Overview of the Solar Radiation Monitoring Laboratory (SRML) Network

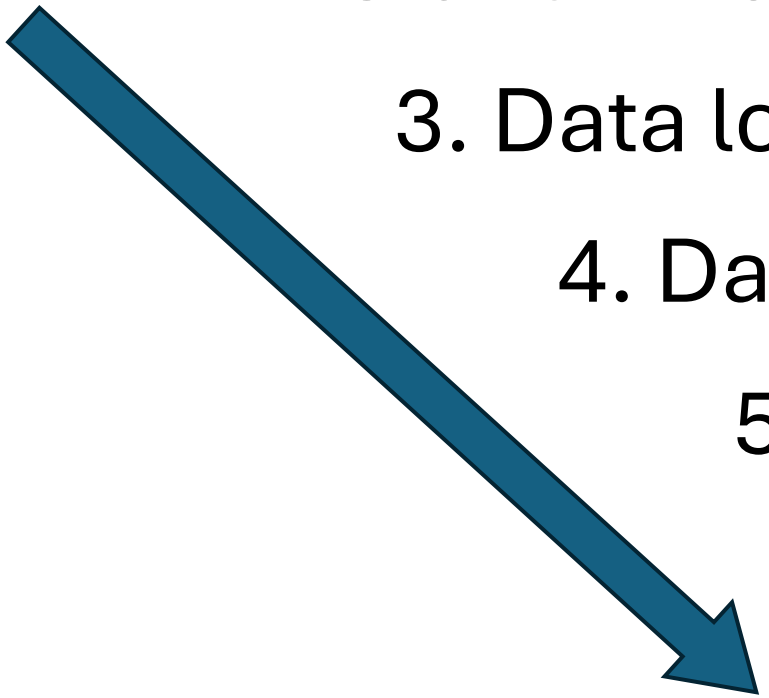
- Mission: generate a long-term, high-quality ground-truth solar resource dataset.
- Operates across Oregon and Washington USA, providing regional solar radiation measurements.
- Established in 1975; currently operates 8 monitoring stations.
- Data is publicly accessible via the SRML website, NREL - MIDC, and pvlib.



https://solardata.uoregon.edu/

Overview of talk

1. Calibrations
2. Station installation
3. Data logger programming
4. Data collection
5. Post processing steps
6. Daily visualization
7. QC protocols



Calibration methods

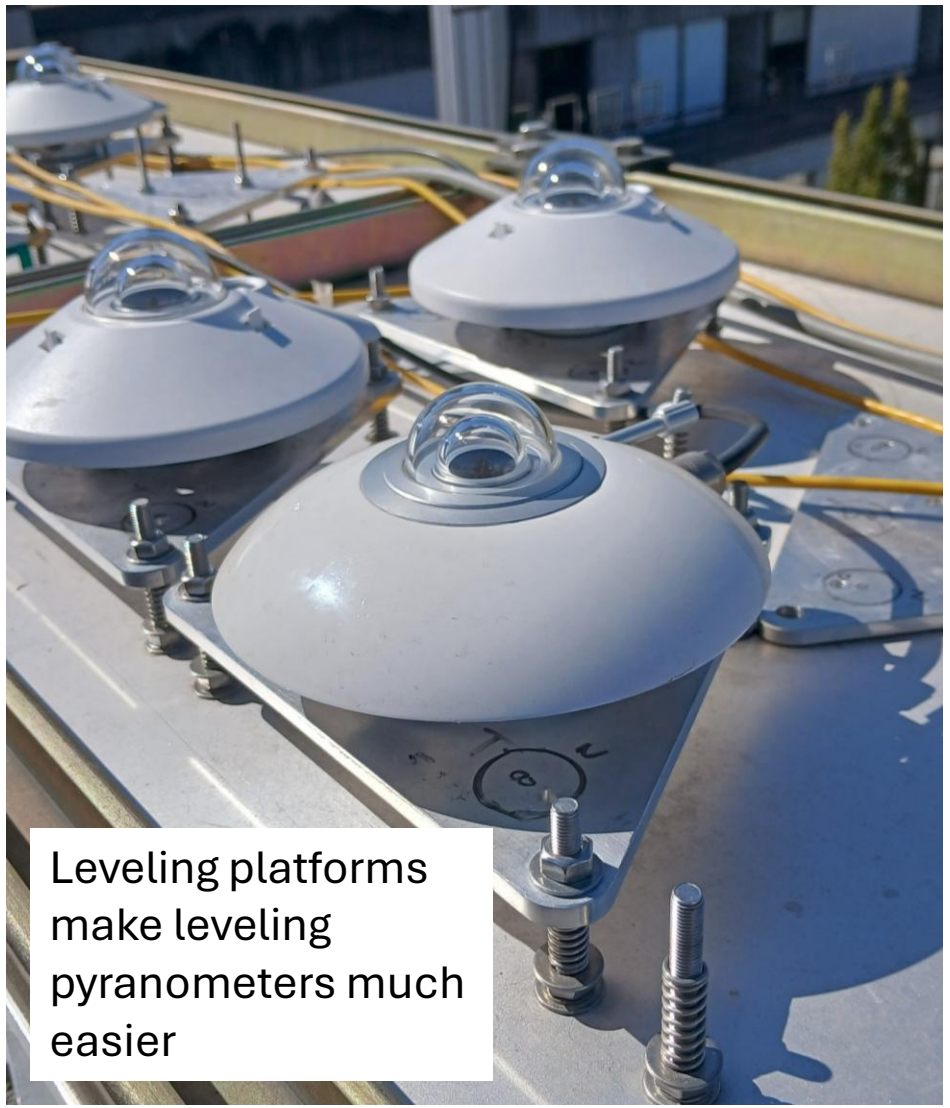
Equations & Definitions:

- Responsivity: $R = \frac{uV}{\text{Ref_Irr}}$
  - R =conversion factor from voltage to irradiance units
  - uV= microvolt signal of sensor
  - Ref_*Irr* = reference irradiance (ACR + calibrated DHI sensor)

Field use:

$$\text{Irr} = \frac{uV}{R}$$

- Converts measured voltage to irradiance in W/m²



Leveling platforms make leveling pyranometers much easier

Calibration methods

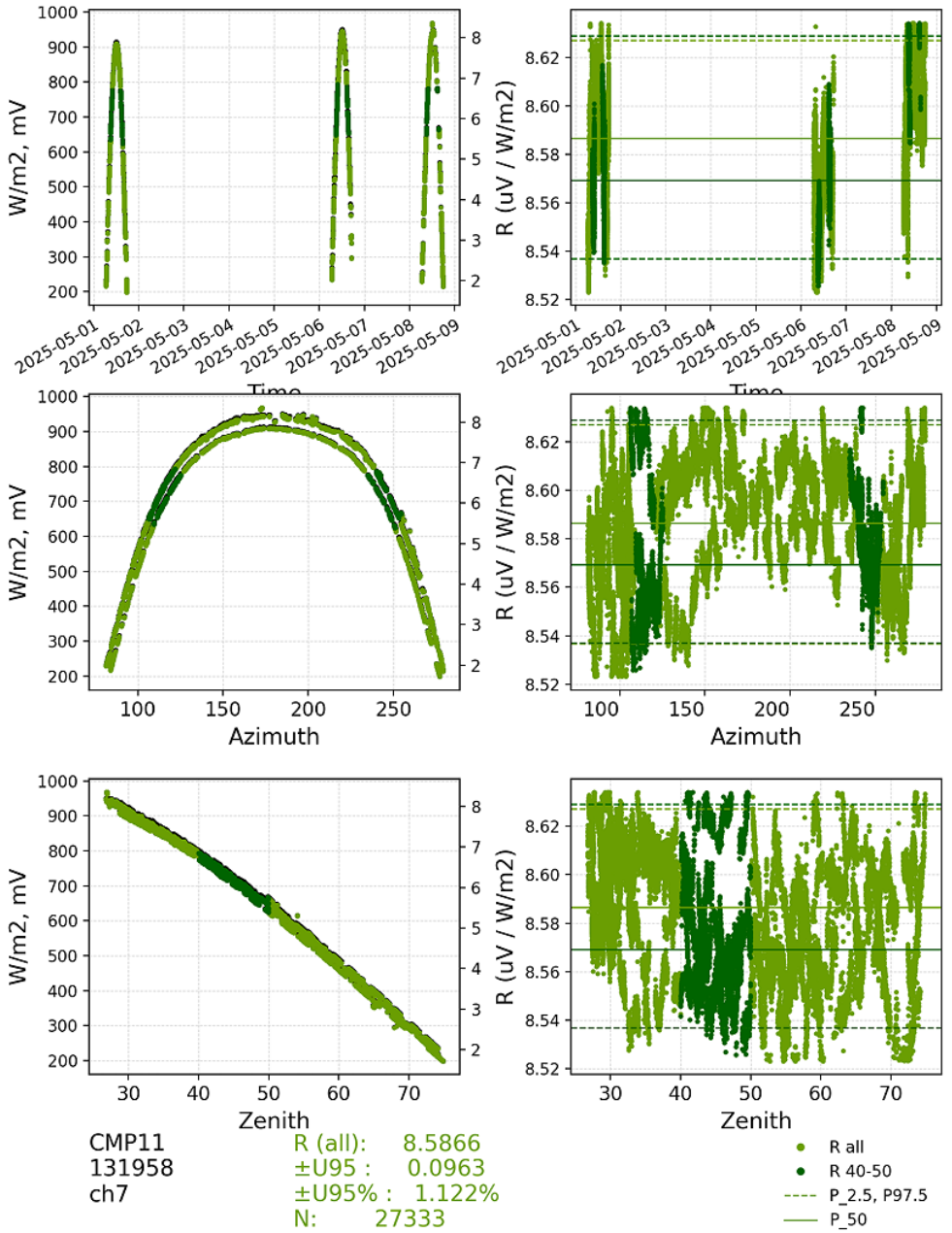
- **Reference & Traceability**
- Uses Eppley Absolute Cavity Radiometer (ACR) as reference for DNI
- Calibrations traceable to WRR via IPC and NPC events

- **A few details**
- Performed outdoors at Eugene station; calibrated sensors redeployed to field
- Typical calibration: 3 full days of data, 2-second sample rate, stable sky conditions

- **Shade/Unshade Calibration**
- Transfers ACR calibration to CMP22 pyranometer (DHI_Ref)
- Timing (example session):
  - 0–2 min: shaded
  - 2–4 min: unshaded (calibration period)
  - 4–17 min: shaded
  - 17–20 min: ACR self calibration

Remove first 40 seconds after each transition

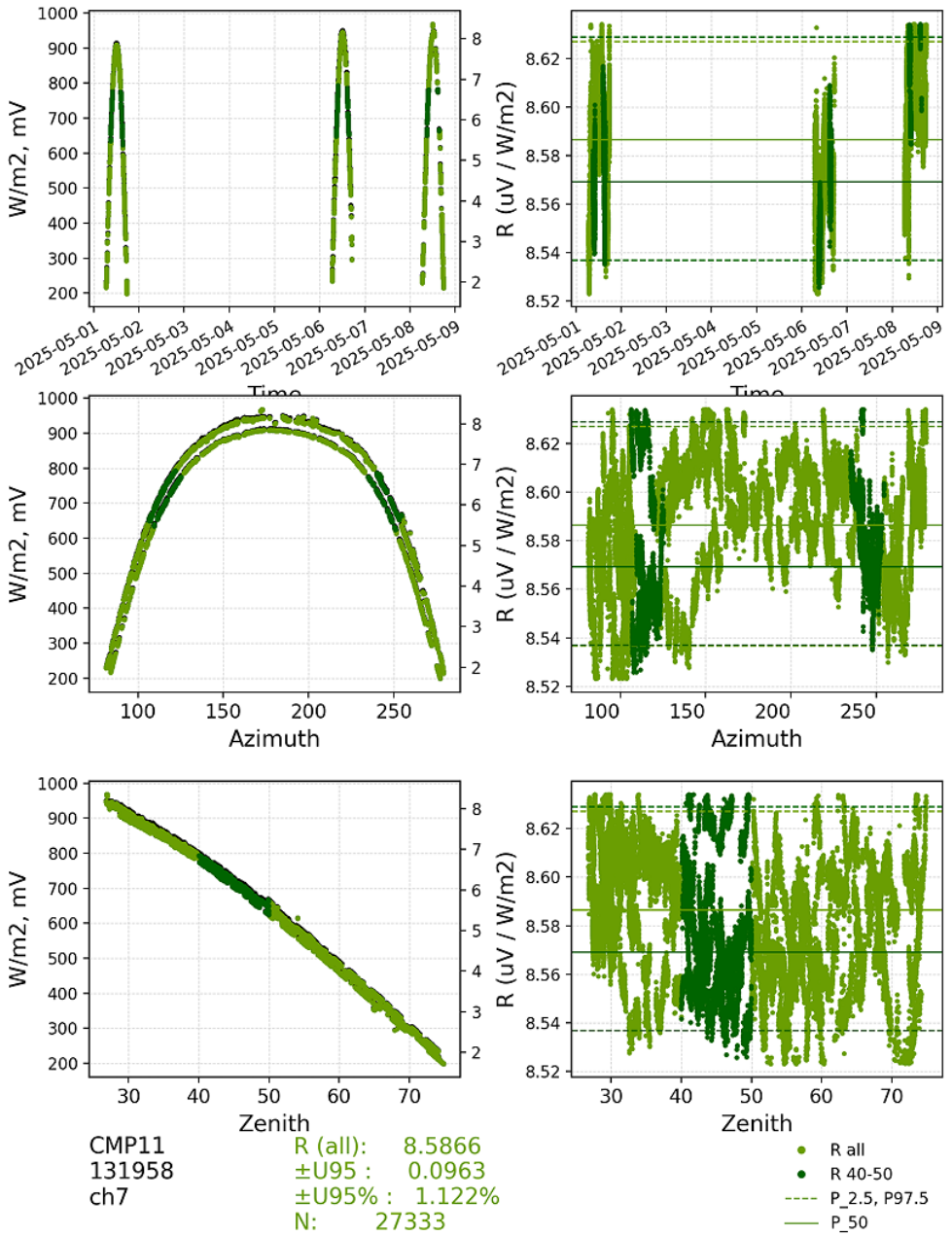
- **Component Sum Calibration**
- Calibrates fleet of sensors from ACR and DHI_Ref



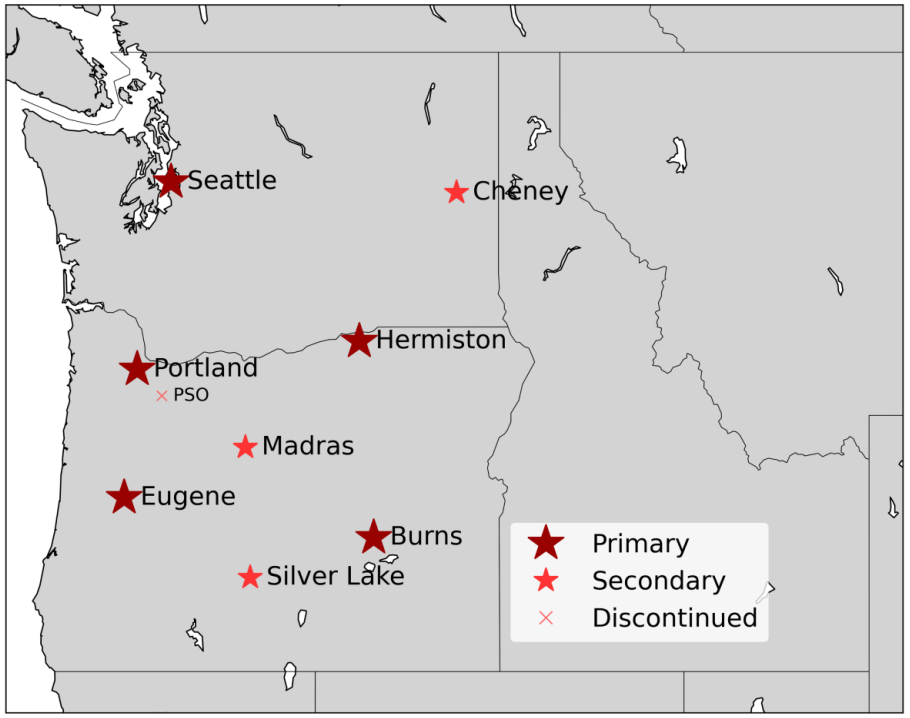
Automated calibration program

- Python script automatically filters outliers and generate responsivity values

1. Load data
2. Compute solar position
3. Remove non-clear sky data
   All sensors
4. Remove sensor specific responsivity data that is not consistent
   Leveling, cleaning, insects …
5. Compute median responsivity



SRML network



5 two-axis tracker stations

Burns, Eugene, Hermiston, Portland, Seattle

- GHI = CMP11 Eugene = CMP22, SR20
- DNI = NIP Eugene = CHP1, DR01-T1
- DHI = CMP11 Eugene = CMP22, SR20

3 RSR secondary stations

Silver Lake, Madras, Cheney

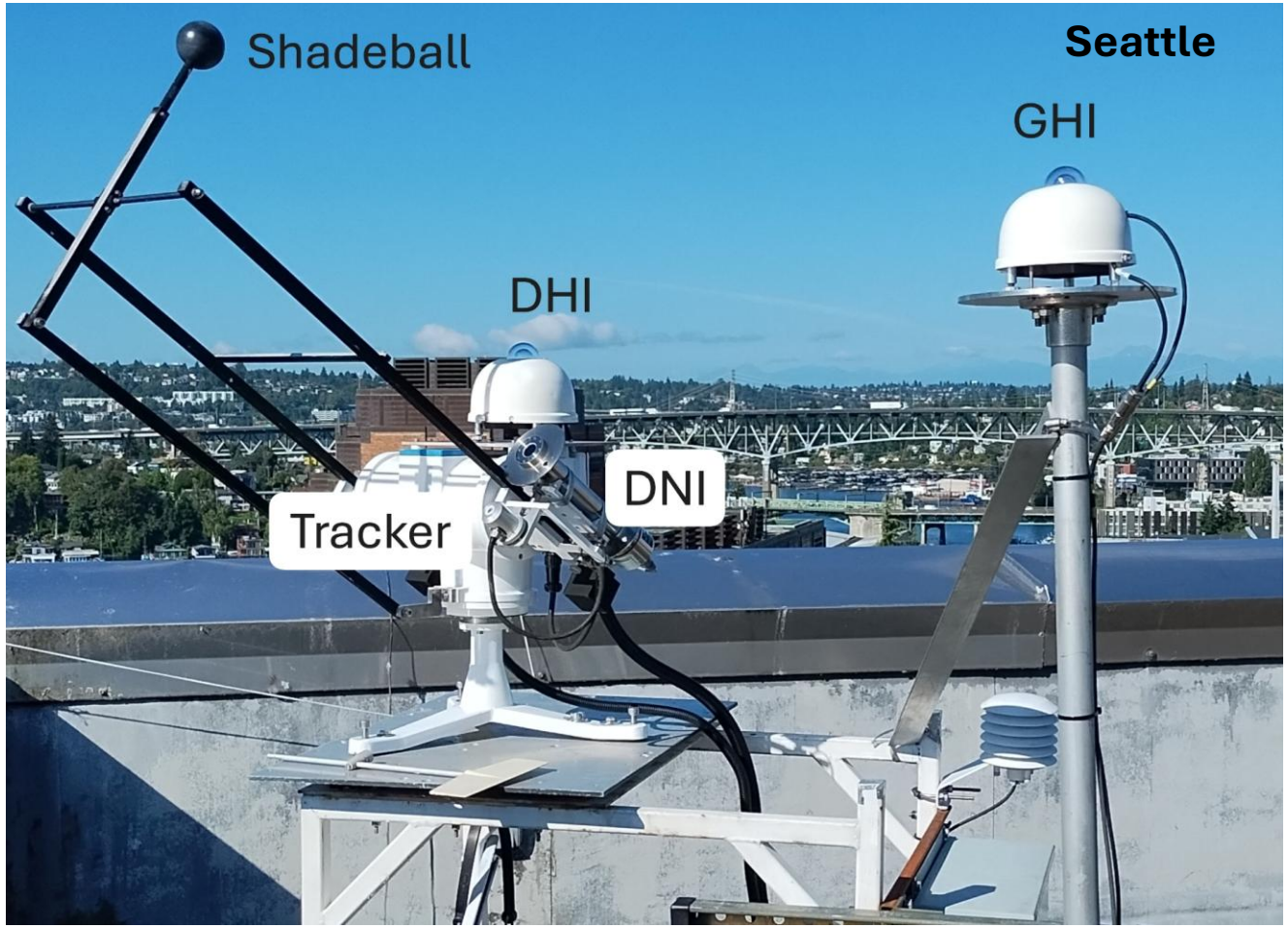
- GHI = CMP11
- GHI, DNI, DHI, = Rotating shadowband radiometer (RSR)

Goal: eventually transition all stations to two-axis tracker stations

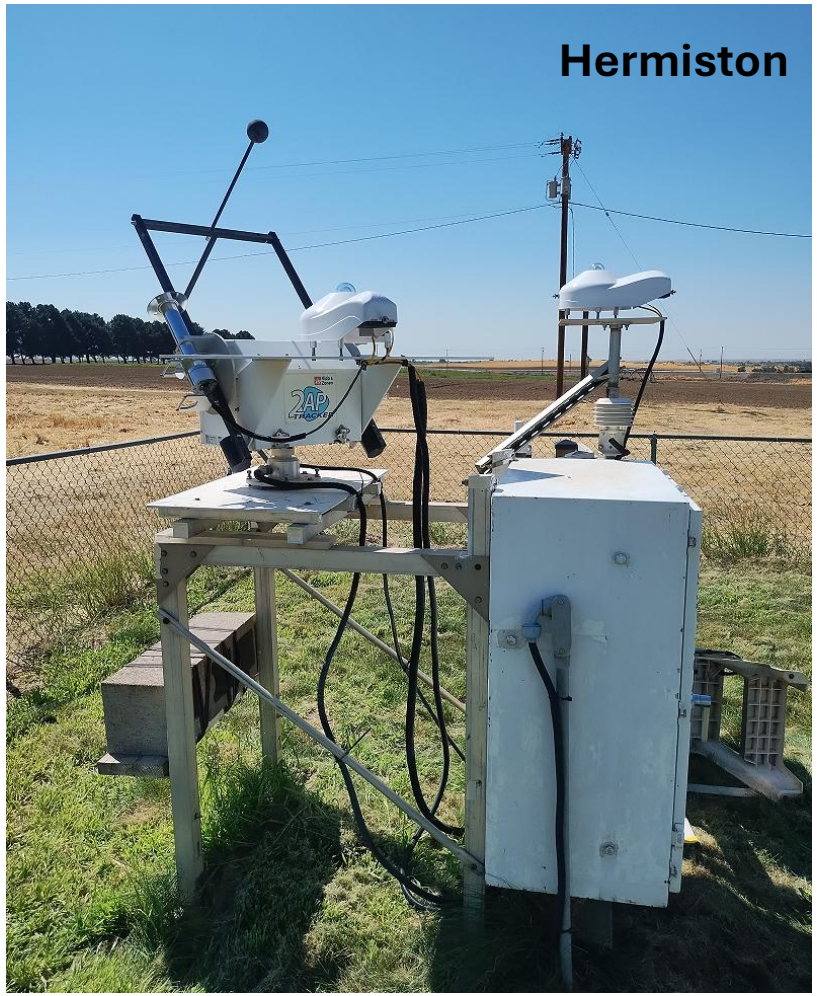
Eugene (EUO) 1975 - 2025

Located on the UO campus.
GHI, DNI, DHI,
Tilted, Downfacing
UV, IR, Spectroradiometer,
MET

Seattle (STW) 2015 – 2025
Portland (PDO) 2024-2025



Hermiston (HEO) 1979 – 2025
Burns (BUO) 1979 -2025

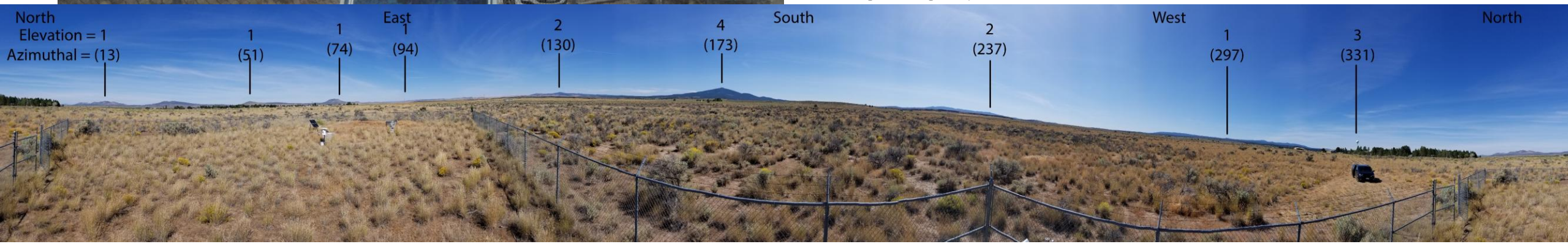


Silver Lake (SIO) 2002 - 2025

Silver lake is representative of the second-class sites.

Oregon can have significant wildfire smoke (this is smoke, not moisture)

360 Panorama views taken at each site. Correlated with measured features on the horizon.



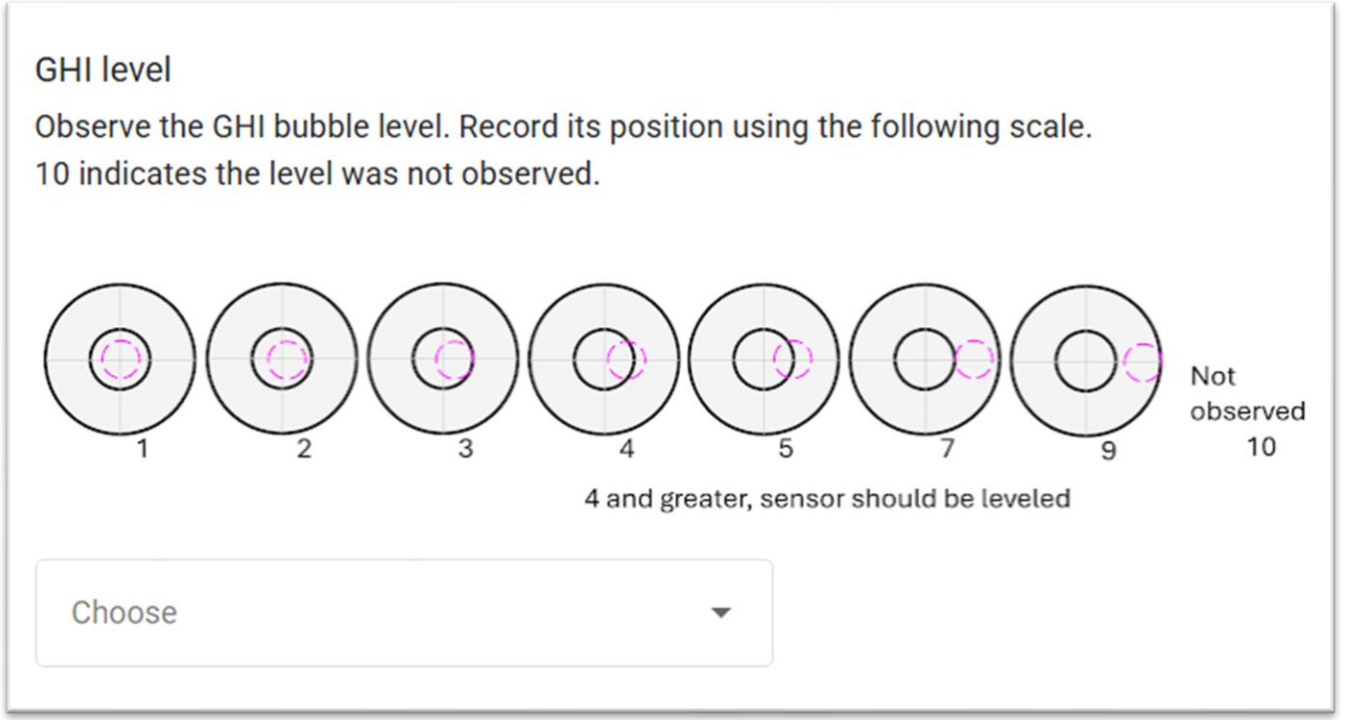
Implemented LT maintenance button push

- When an local technician (LT) visits a site, they press a maintenance button.
- The button press is logged in the output table.
- This marker is used for QC, since maintenance often causes bad data

```
412     ' Detect if the button was pushed. (a pulse)
413     PulseCount (maint_button_push,1,14,2,0,1.0,0)
414
415     ' If detected
416     If maint_button_push > 0 Then
417       ' Only get in here if button is pushed
418       maint_timer = 0
419       maint_event = 1
420       maint_button_push = 0
421     EndIf
422
423     ' Increase timer in seconds
424     maint_timer = maint_timer + Scan_c
425     maint_timer_day = maint_timer / 86400
426
427     ' if the count is 5 minutes (100 3 second scans)(1
428     If maint_timer > 300 Then
429       maint_event = 0
430     EndIf
```

Online maintenance log sheets

- Use Google forms to generate an online maintenance form
- Goal: allows LT to report station status to the website in real time.
- LTs can use their phone to report values



	A	B	C	D	E
1	Date_Time (LST)	Conditions	Conditions	Conditions	Cond
2	2025-07-08_15:10	No	No	No	No
3	2025-07-12_06:25	No	No	No	No
4	2025-07-14_06:10	No	No	No	No
5	2025-07-15_05:28	No	No	No	No
6	2025-07-17_05:37	No	Yes - a bit	No	No
7	2025-07-28_16:23	No	Yes - a bit	No	No
8	2025-07-31_13:06	No	No	No	No
9	2025-08-04_12:43	No	No	No	No

Data loggers

- Campbell
  - CR1000, 1000x, 3000, 6
- Scan rate 1, 2, 3 seconds (depends on station)
- Minute average table

- Measure voltage of sensors
- Apply responsivity on logger to generate irradiance
- Put both mV and Irr in output tables

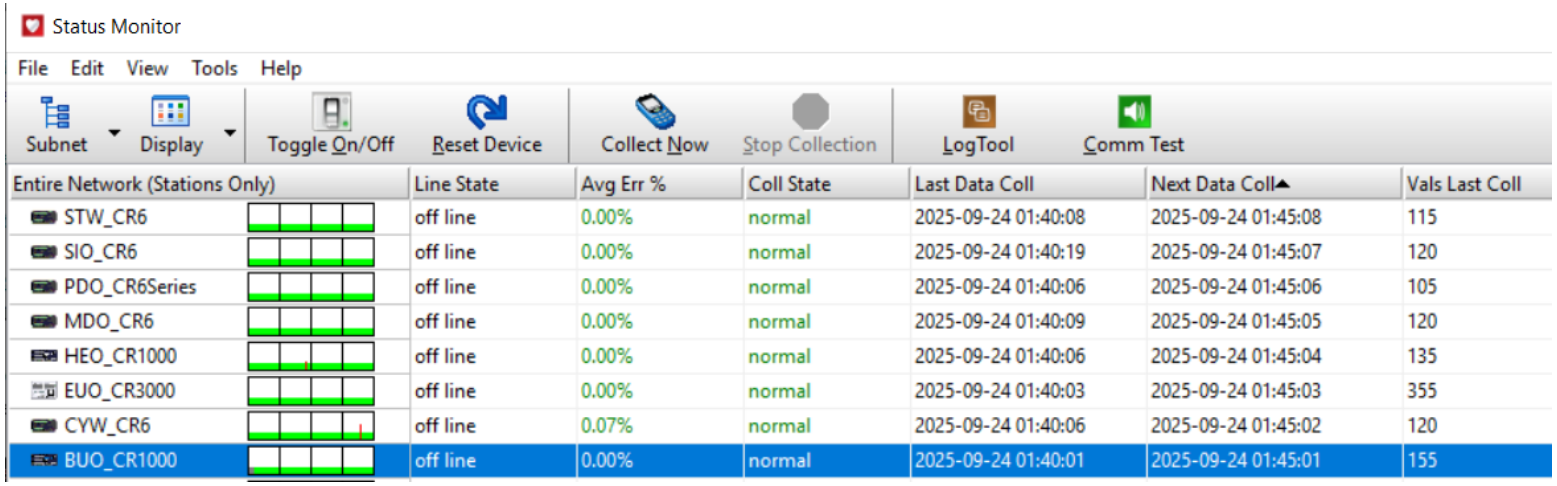
```
463     ' GHI measurements
464     VoltDiff(GHI_mV,1,mV25,3,True,0,_60Hz,1.0,0.0)
465     GHI_Irr = GHI_mV * 1000 / GHI_R
466
467     ' DNI measurements
468     VoltDiff (DNI_mV,1,mV25,2,True,0,_60Hz,1,0)
469     DNI_Irr = DNI_mV * 1000 / DNI_R
470
471     ' DHI measurements
472     VoltDiff (DHI_mV,1,mV25,4,True,0,_60Hz,1,0)
473     DHI_Irr = DHI_mV * 1000 / DHI_R
```

Also computed on logger:

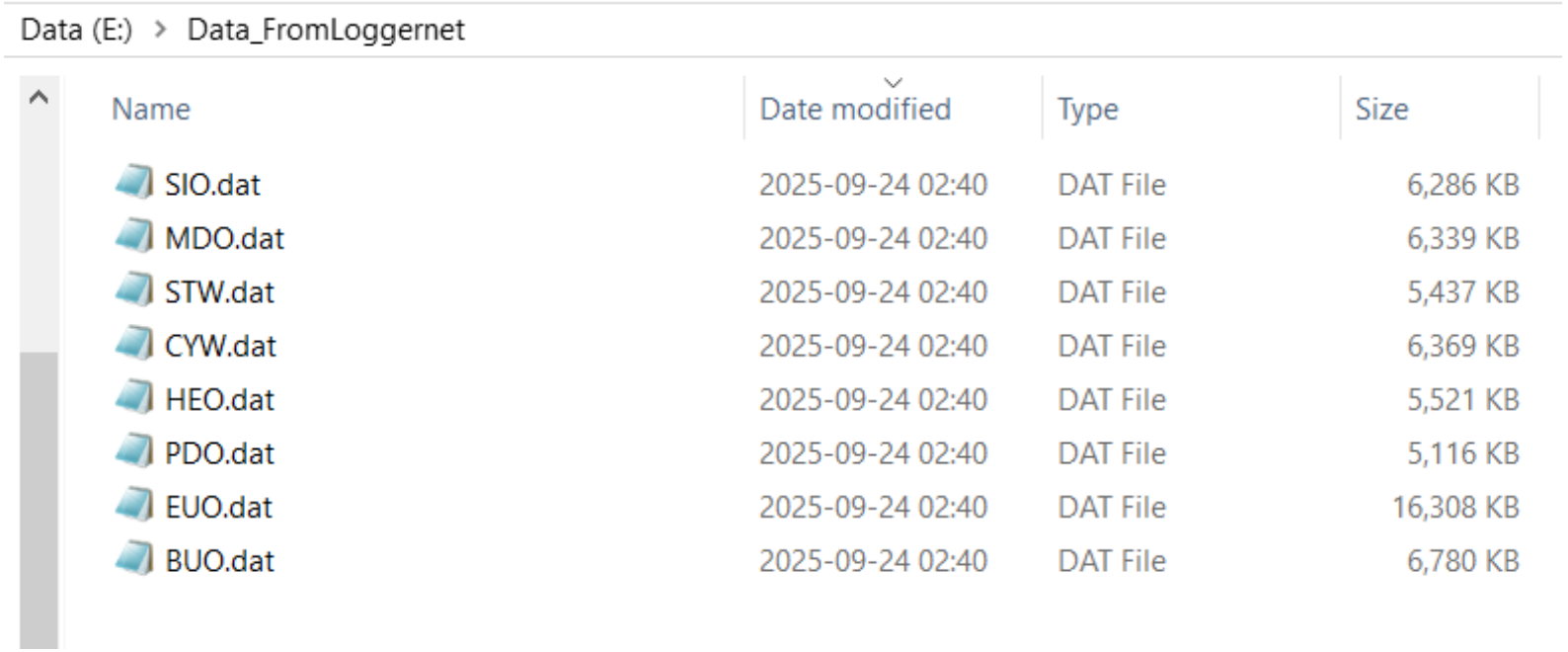
- Solar position
- Percent difference between GHI and GHI_Calc
- Maintenance button push recorded
- Set 2AP tracker clock with logger.
- Time it takes to complete scan

Data collection via Loggernet

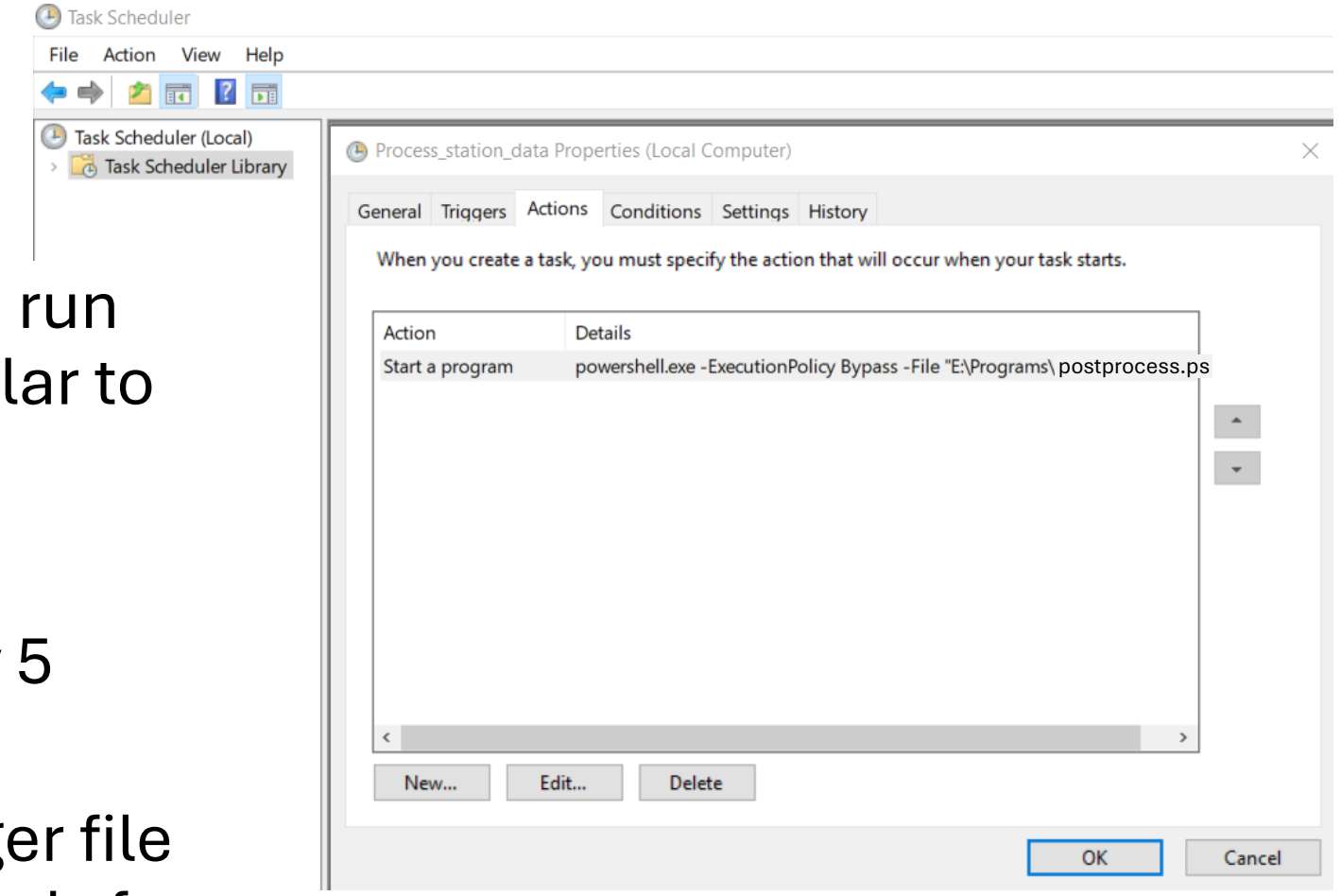
- Data collected every 5 minutes
- Collected via cell phone or direct internet connection
- Data also collected by NREL-MIDC



Entire Network (Stations Only)	Line State	Avg Err %	Coll State	Last Data Coll	Next Data Coll	Vals Last Coll
STW_CR6	off line	0.00%	normal	2025-09-24 01:40:08	2025-09-24 01:45:08	115
SIO_CR6	off line	0.00%	normal	2025-09-24 01:40:19	2025-09-24 01:45:07	120
PDO_CR6Series	off line	0.00%	normal	2025-09-24 01:40:06	2025-09-24 01:45:06	105
MDO_CR6	off line	0.00%	normal	2025-09-24 01:40:09	2025-09-24 01:45:05	120
HEO_CR1000	off line	0.00%	normal	2025-09-24 01:40:06	2025-09-24 01:45:04	135
EUO_CR3000	off line	0.00%	normal	2025-09-24 01:40:03	2025-09-24 01:45:03	355
CYW_CR6	off line	0.07%	normal	2025-09-24 01:40:06	2025-09-24 01:45:02	120
BUO_CR1000	off line	0.00%	normal	2025-09-24 01:40:01	2025-09-24 01:45:01	155



Post processing – near real time



- Microsoft's Task Scheduler to run tasks every five minutes (similar to Loggernet Task Manager)

- PowerShell script is run every 5 minutes
- Goal: take data from .dat logger file and generate useable files ready for public dissemination

Post processing – near real time

Powershell code

Goes through each station

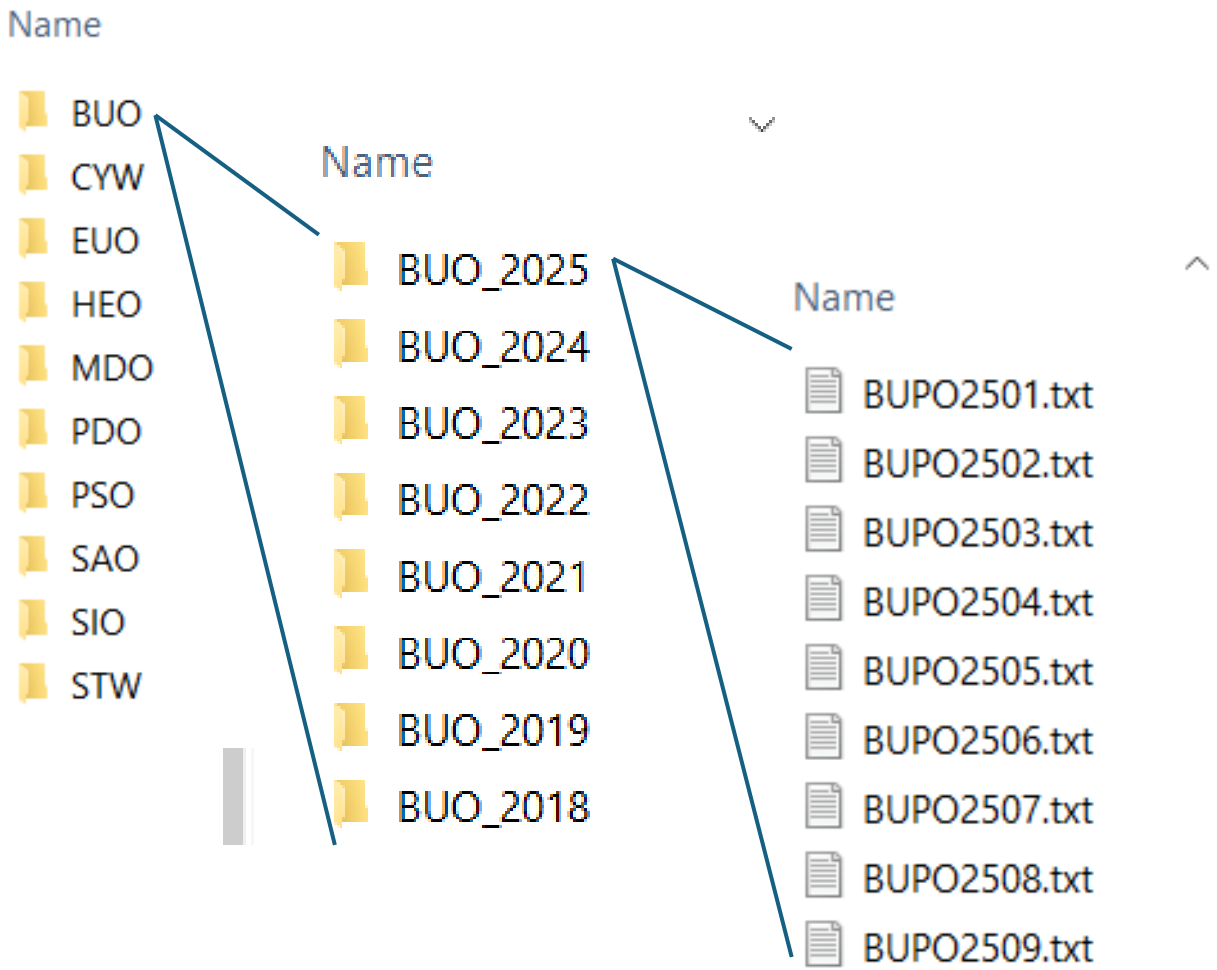
Calls various python scripts

Each python script performs tasks to data

```
# Run all the processing scripts
python E:\                    Address \SRML_AddMeasuredData.py $datLocat
python E:\                    Address \SRML_DailySummary.py $cfLocationP
python E:\                    Address \SRML_AddAdjustedData.py $cfLocati
python E:\                    Address \SRML_DailySum.py $cfLocationPy
python E:\                    Address \SRML_MakePreviousFormat.py $sn3 $
python E:\                    Address \SRML_ZipFiles.py $sn3
python E:\                    Address \SRML_MakePlots.py $sn3 $cfLocatio
```

Data storage structure

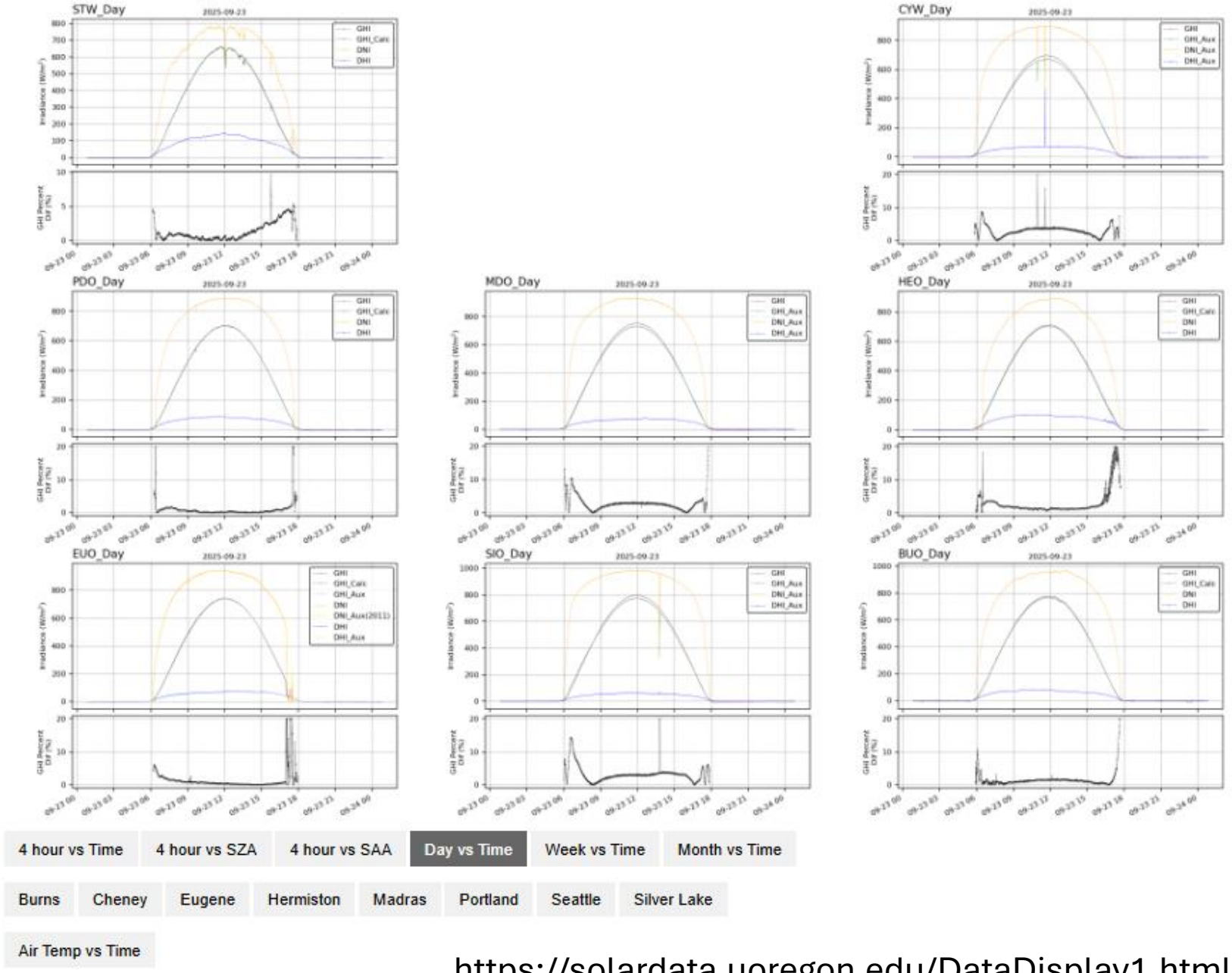
- The SRML stores data in .txt and csv files
- Does not use a formal database
- The SRML stores data in one month station files
- Station / Year / Monthly file



Daily visualizations to the website

- Final post processing step generates current images of data.
- Published to the web
- Static .png image names referenced by html.

- NREL – SRRL's dashboard is relatively easy because they only have one location to show.

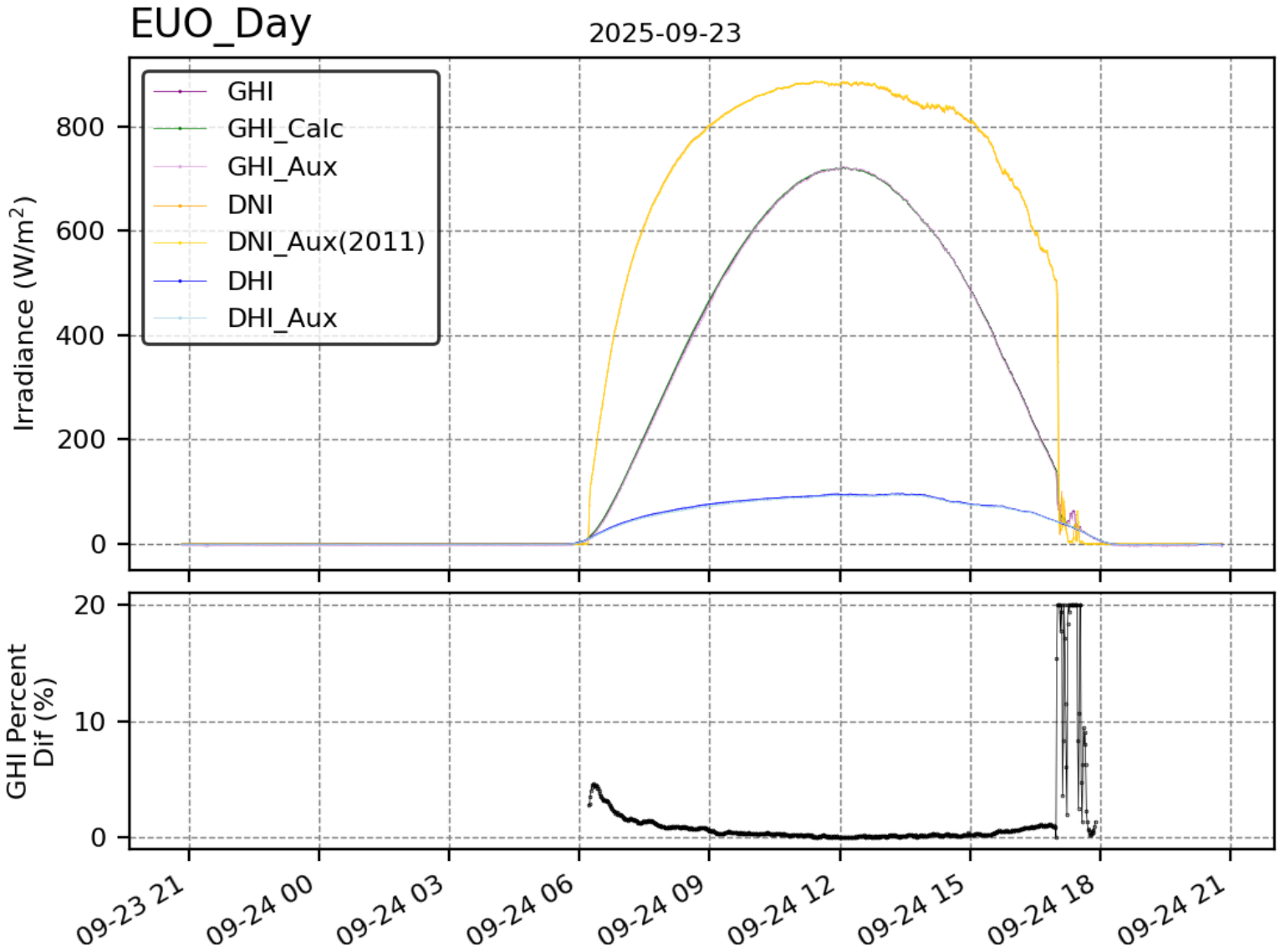


https://solardata.uoregon.edu/DataDisplay1.html

Daily visualizations to the website

- Final post processing step generates current images of data.
- Published to the web
- Static .png image names referenced by html.

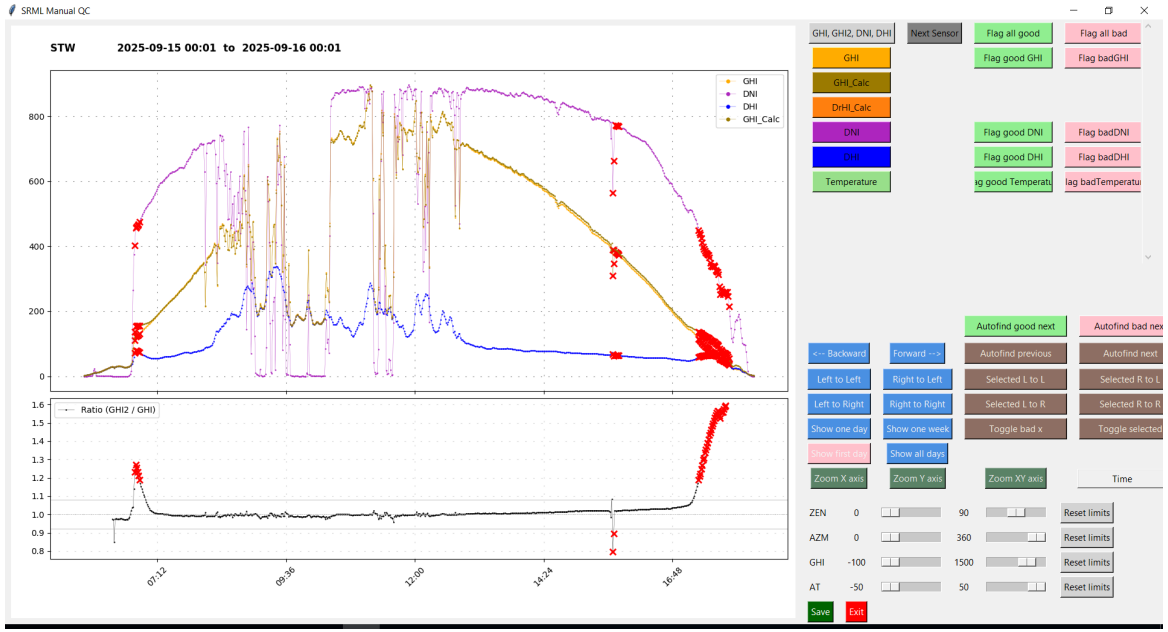
- NREL – SRRL's dashboard is relatively easy because they only have one location to show.



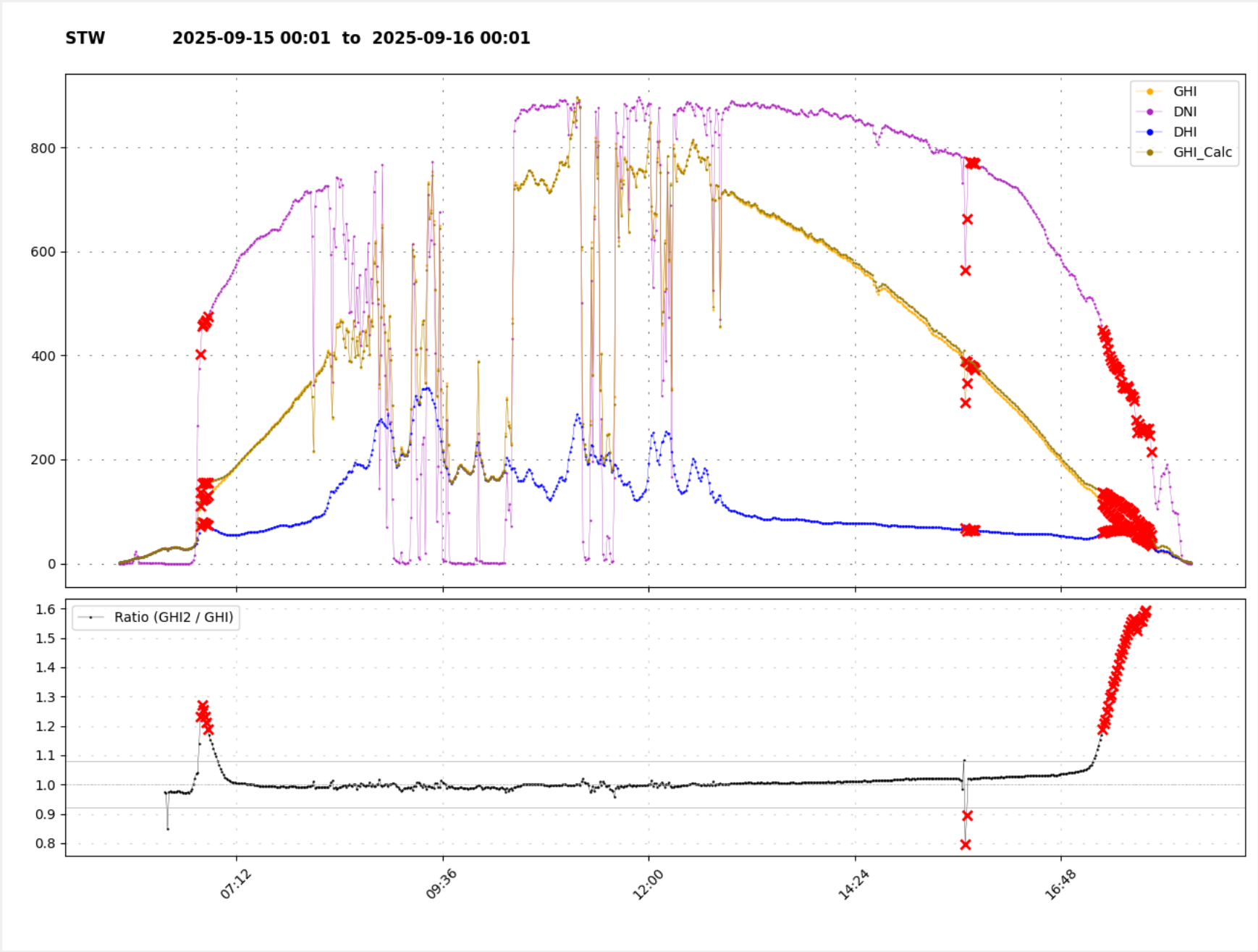
https://solardata.uoregon.edu/DataDisplay1.html

SRML quality control (QC) process

- At the end of each month, files are QC'd.
- Goal:
  - flag good data good
  - flag bad data bad
- First automated QC implemented. Based on BSRN QC.
- Then manual QC applied via GUI.
  - Many buttons and features.
  - GUI based in python, uses Tkinter



SRML Manual QC



GHI, GHI2, DNI, DHI | Next Sensor | Flag all good | Flag all bad

GHI | Flag good GHI | Flag badGHI

GHI_Calc

DrHI_Calc

DNI | Flag good DNI | Flag badDNI

DHI | Flag good DHI | Flag badDHI

Temperature | g good Temperatu | lag badTemperatu

Autofind good next | Autofind bad nex

<-- Backward | Forward --> | Autofind previous | Autofind next

Left to Left | Right to Left | Selected L to L | Selected R to L

Left to Right | Right to Right | Selected L to R | Selected R to R

Show one day | Show one week | Toggle bad x | Toggle selected

Show first day | Show all days

Zoom X axis | Zoom Y axis | Zoom XY axis | Time

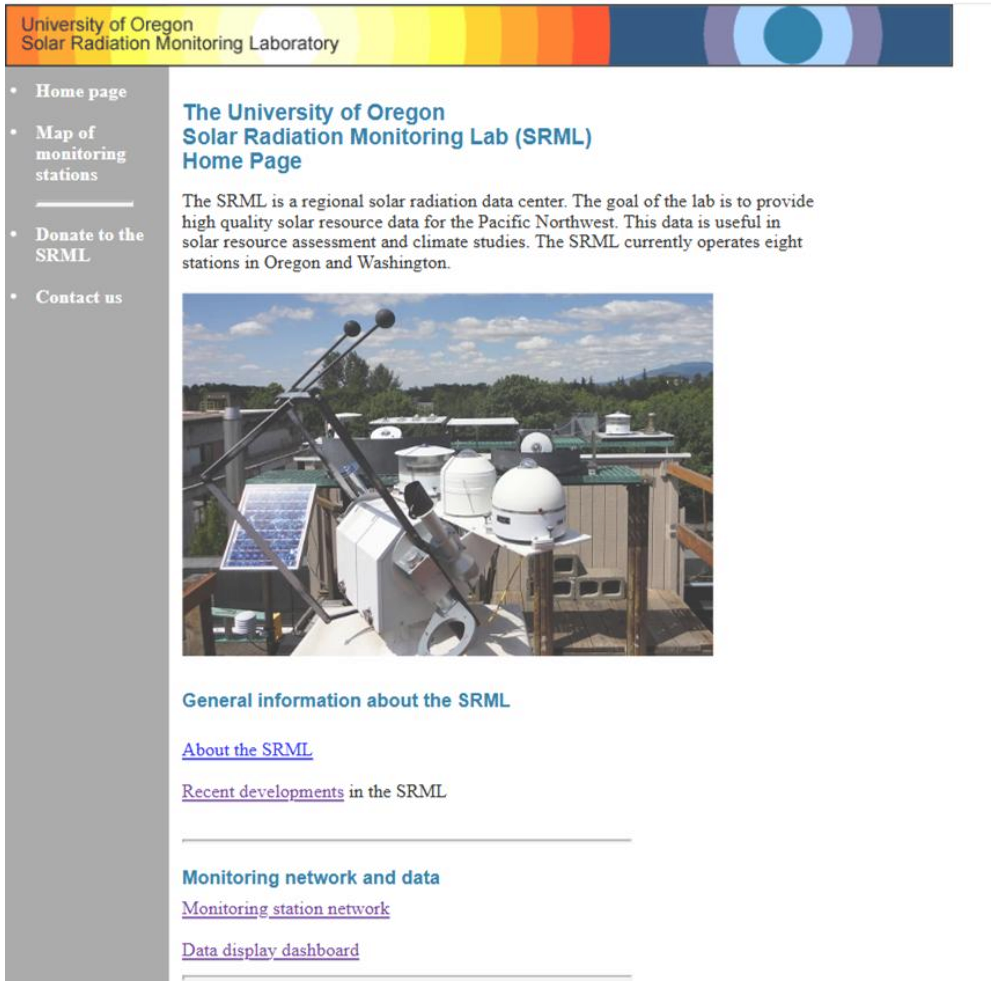
ZEN	0	90		Reset limits
AZM	0	360		Reset limits
GHI	-100	1500		Reset limits
AT	-50	50		Reset limits

Save | Exit

SRML Manual QC
STW 2025-09-15 14:59 to 2025-09-15 15:55
Select desired range
Flag data accordingly
GHI
GHI_Calc
DrHI_Calc
DNI
DHI
Ratio (GHI2 / GHI)
GHI, GHI2, DNI, DHI
Next Sensor
Flag all good
Flag all bad
GHI
Flag good GHI
Flag badGHI
GHI_Calc
DrHI_Calc
DNI
Flag good DNI
Flag badDNI
DHI
Flag good DHI
Flag badDHI
Temperature
Autofind good next
Autofind bad ne
<-- Backward
Forward -->
Autofind previous
Autofind next
Left to Left
Right to Left
Selected L to L
Selected R to L
Left to Right
Right to Right
Selected L to R
Selected R to R
Show one day
Show one week
Toggle bad x
Toggle selecte
Show first day
Show all days
Zoom X axis
Zoom Y axis
Zoom XY axis
Time
ZEN 0 90 Reset limits
AZM 0 360 Reset limits
GHI -100 1500 Reset limits
AT -50 50 Reset limits
Save
Exit

Comments / Questions

- Hopefully, you found some topics useful.
- Let me know of ways that I can do things better.
- Find me during the IPC to discuss methodologies in greater detail



Josh Peterson
jpeters4@uoregon.edu
https://solardata.uoregon.edu/