PLANTPREDICT UPDATE & SDK DEVELOPMENT FOR API

Stephen Kaplan
Performance & Prediction Engineer

2018 PVPM Conference

First Solar.

LEADING THE WORLD'S SUSTAINABLE ENERGY FUTURE

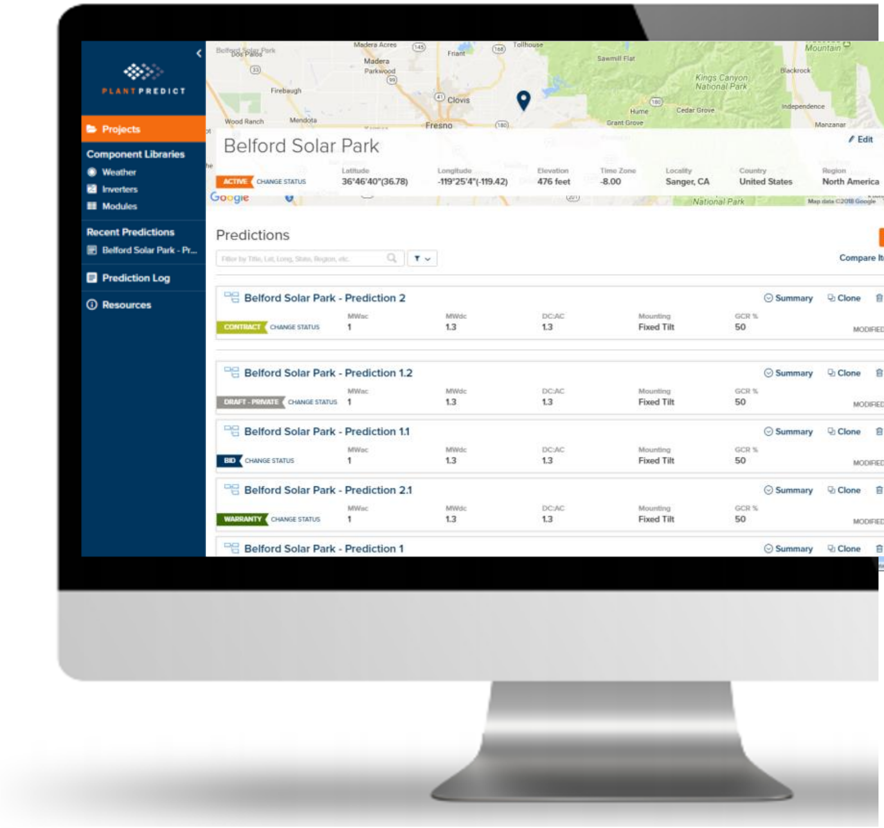
NOTICE TO RECIPIENTS

The Recipient acknowledges that (i) this presentation is provided to the Recipient for informational purposes only, (ii) neither First Solar, Inc. (“First Solar”) nor any of its affiliates bears any responsibility (and shall not be liable) for the accuracy or completeness (or lack thereof) of any information contained within this presentation and the oral statements made in connection therewith and this presentation and the oral statements made in connection therewith are not to be relied upon as a substitute for independent review of such other information as the Recipient may deem appropriate or prudent to review, (iii) no representation regarding the material within this presentation and the oral statements made in connection therewith is made by First Solar or its affiliates, (iv) neither First Solar nor any of its affiliates have made an independent verification as to the accuracy or completeness of the information within this presentation and the oral statements made in connection therewith, and (v) neither First Solar nor any of its affiliates shall have any obligation to update or supplement any information contained within this presentation and the oral statements made in connection therewith or otherwise provide additional information.

PLANTPREDICT OVERVIEW

INTRODUCING PLANTPREDICT

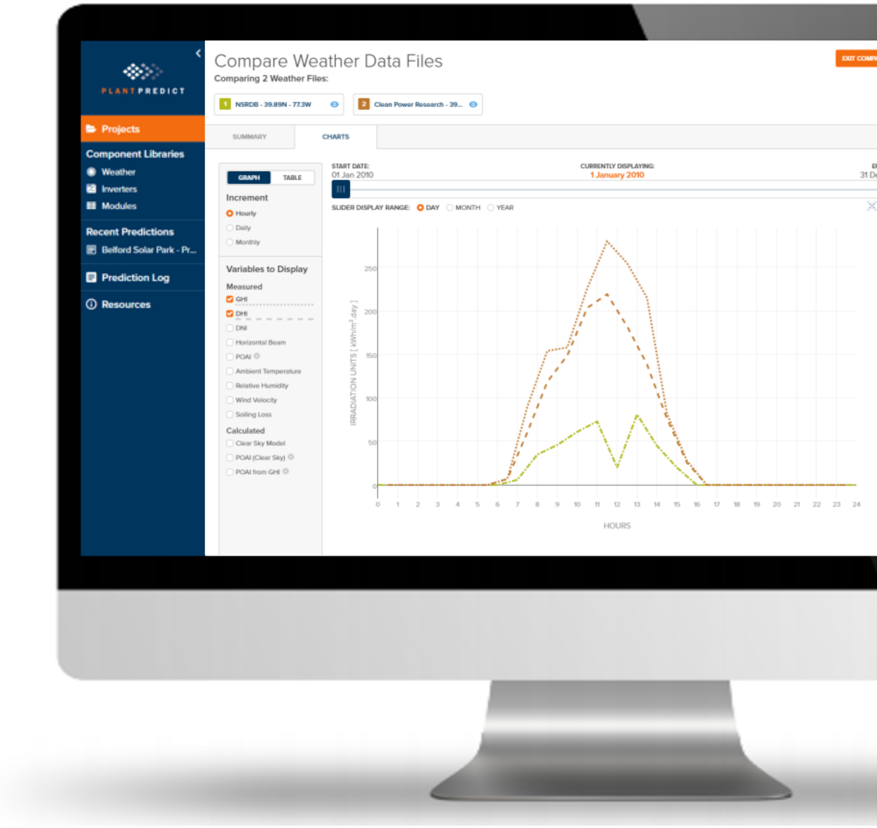
- Designed with the **utility-scale** PV industry in mind
- Free
- Web-based tool
- Reduce prediction time by up to 75%
- Used in over 350 MWAC of contracted utility-scale PV projects
- All algorithms documented and published on www.plantpredict.com
- Independently reviewed and benchmarked against over 1 GW of operating facilities



Reviewed by: DNV·GL, RV, leidos, ICF

PLANTPREDICT FEATURES

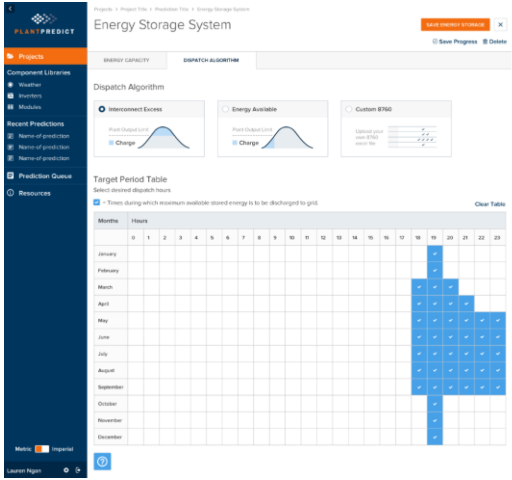
- *End-to-end utility-scale modeling*
  - — Sub-hourly and multi-year predictions
  - — Built-in spectral correction
  - — Built-in MV and HV transformers
  - — Built-in Availability and LGIA losses
  - — No need for pre- or post-processing
- *Everything in one tool*
  - — One-click weather download from leading providers
  - — Pre-loaded with industry-standard weather, module, and inverter files
- *Advanced features, simple interface*
  - — Compare weather and predictions side-by-side
  - — Optimize your design with Batch, Clone, and Quick Edit
  - — Cloud-hosted for ease of sharing and data security
  - — API available for automation

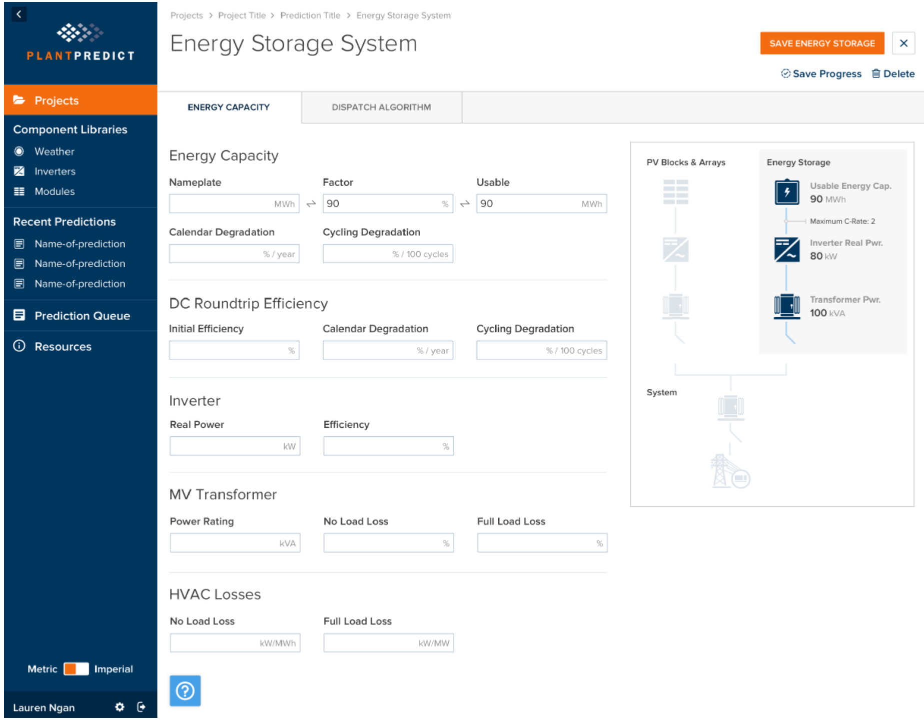


UPCOMING FEATURES

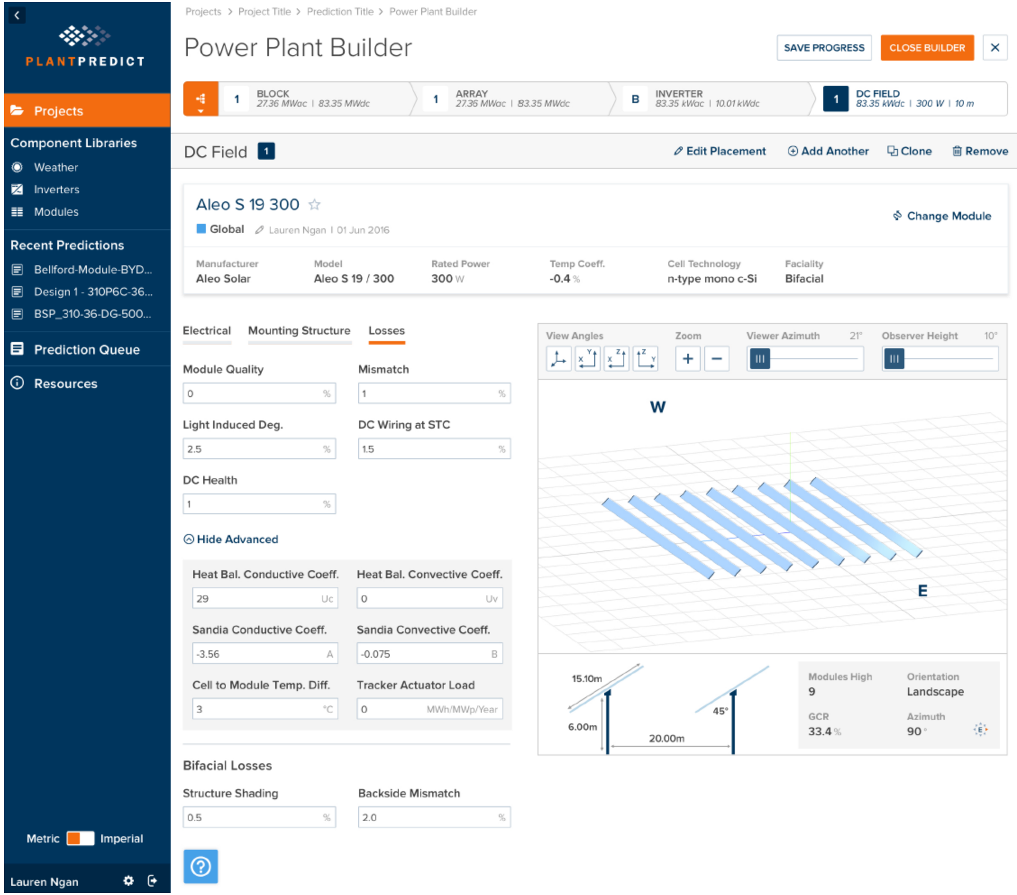
PV Energy Storage | Bifacial PV | Module File Creator

- *3 modeling options*
  1. **LGIA Excess** energy used to charge battery. Energy is discharged during user-defined target period.
  2. **Energy Available** is used to charge the battery until it is full. Energy is discharged during user-defined target period.
  3. **Custom** option allows user to define both charging and discharging.



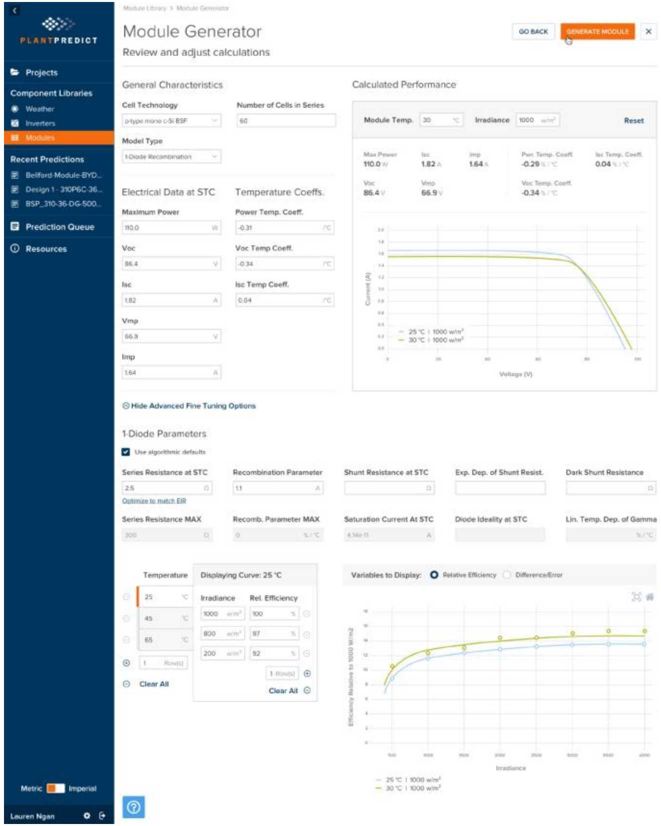


- Characterize modules as “Monofacial” or “Bifacial”.
- Model structure shading and backside mismatch and post height for a DC field.
- Model transmission and bifaciality for a PV module.
- Uses the NREL 2D View Factor model (Marion)

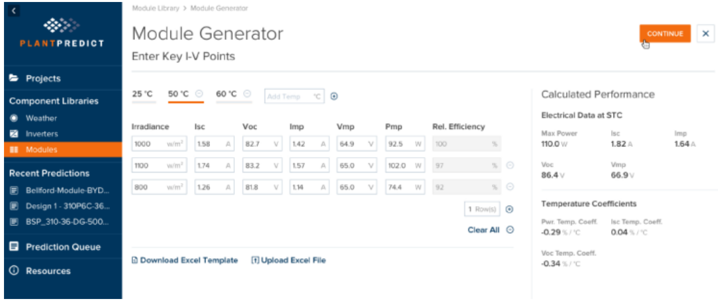


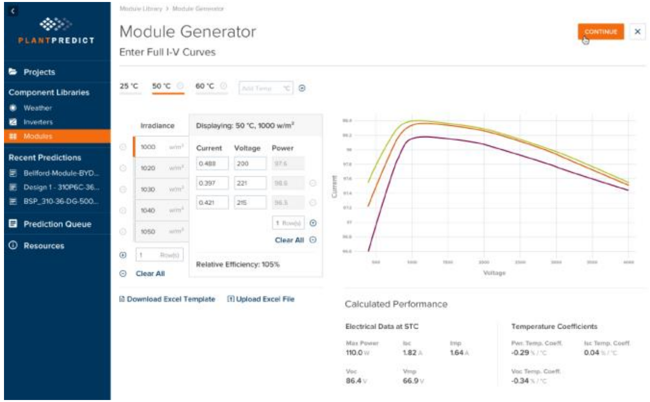
Module File Creator

Calculate single-diode parameters from datasheet parameters and adjust to match a target effective irradiance response.



Input data via IEC61853-type grid or full I-V curves.





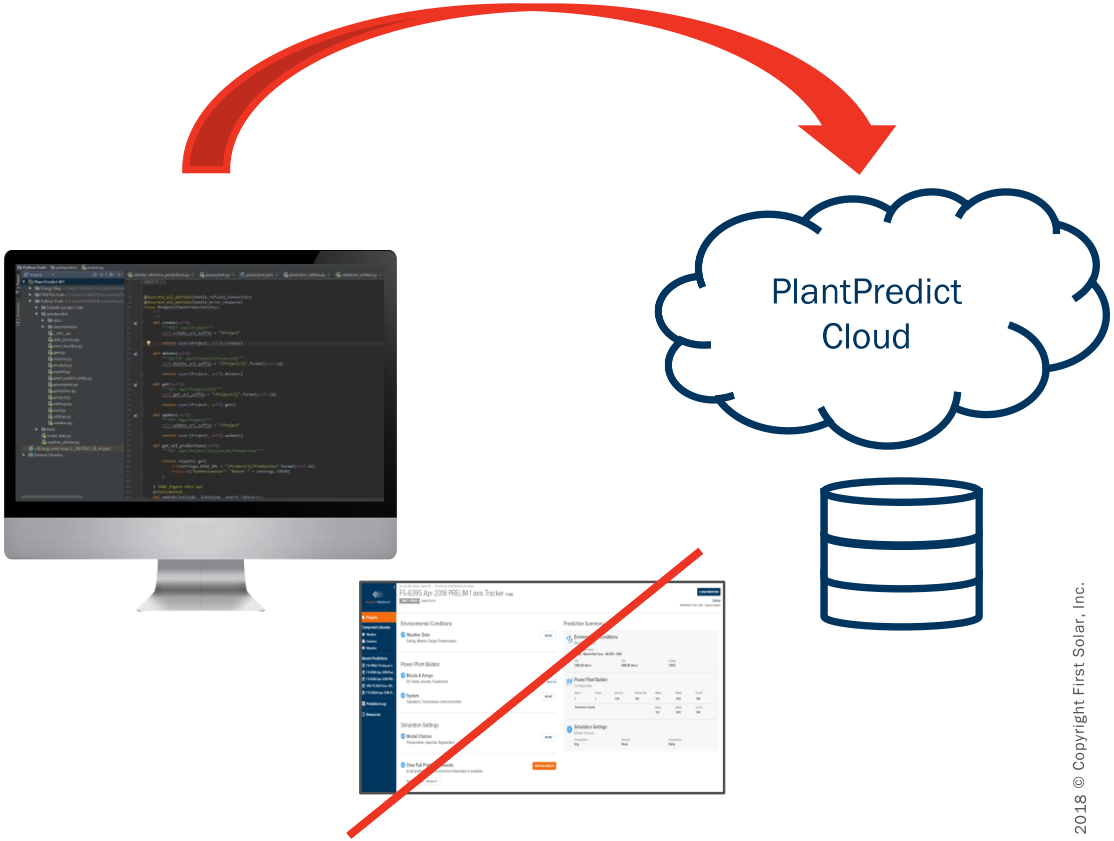
PLANTPREDICT API & CASE STUDY

PlantPredict API

- ***PlantPredict's Applied Programming Interface (API) allows developer access to broad functionality.***

- ***Automate everything***
  - Build power plants, run predictions, upload weather data, and more...

- ***Endless applications***
  - Business/O&M software integration
  - Run batches of predictions with control over *every parameter*
  - Utilize as a weather/module/inverter database for any external application



Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.

Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.

python™

Run all prediction variants in a project.

```python
# Run all predictions in a project.
import plantpredict


project = plantpredict.Project(id=1)
for p in project.get_all_predictions():
    prediction = plantpredict.Prediction(id=p['id'], project_id=project.id)
    prediction.run()
```

Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.

Run all prediction variants in a project.

```python
# Run all predictions in a project.
import plantpredict


project = plantpredict.Project(id=1)
for p in project.get_all_predictions():
    prediction = plantpredict.Prediction(id=p['id'], project_id=project.id)
    prediction.run()
```

Download and retrieve a weather file at a specified location..

```python
# Download and retrieve a weather file from Meteonorm.
import plantpredict


weather = plantpredict.Weather()
weather.download(latitude=30, longitude=-110, provider=plantpredict.WEATHER_DATA_PROVIDER['METEONORM'])
weather.get()
```

Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.

Run all prediction variants in a project.

```
# Run all predictions in a project.
import plantpredict


project = plantpredict.Project(id=1)
for p in project.get_all_predictions():
    prediction = plantpredict.Prediction(id=p['id'], project_id=project.id)
    prediction.run()
```

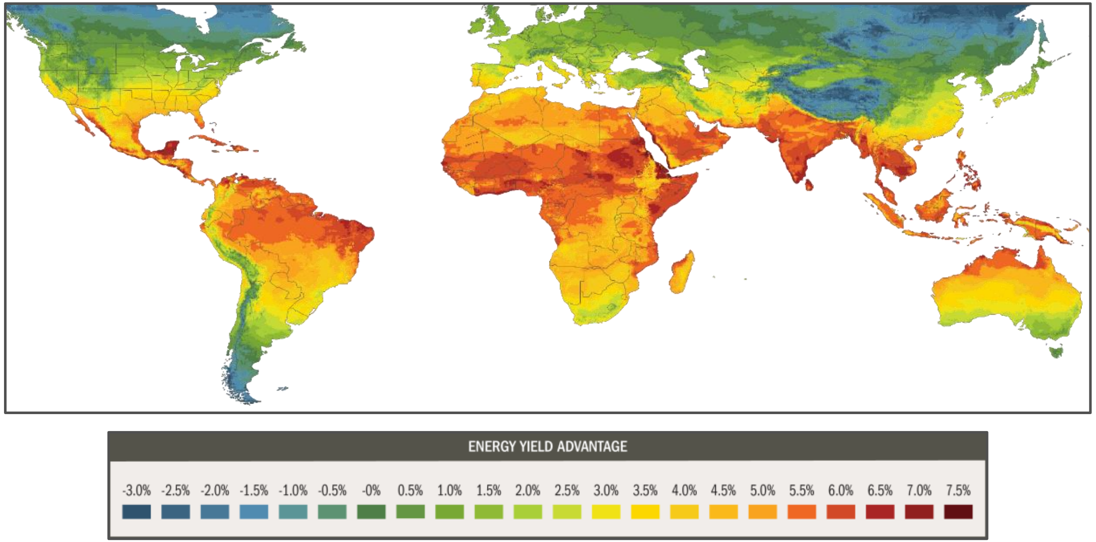
Update module used in energy prediction.

```
# Update module used in simulation.
import plantpredict


power_plant = plantpredict.PowerPlant(project_id=1, prediction_id=2)
power_plant.get()
power_plant.blocks[0]['arrays'][0]['inverters'][0]['dcFields'][0]['moduleId'] = 4
power_plant.update()
```

Download and retrieve a weather file at a specified location..

```
# Download and retrieve a weather file from Meteonorm.
import plantpredict


weather = plantpredict.Weather()
weather.download(latitude=30, longitude=-110, provider=plantpredict.WEATHER_DATA_PROVIDER['METEONORM'])
weather.get()
```

Case Study – Global Module Energy Yield Comparison



Integrated the PlantPredict Python SDK with Python statistics and plotting libraries to generate a global heat map, representing the relative performance of 2 modules.

STEPS

1. Automatically configure and run predictions at 100 representative locations across the globe for 2 modules.
2. Run a regression to correlate weather data with relative energy production.
3. Use a global grid of annual weather data at 0.25º (latitude) x 0.25º (longitude) resolution to calculate relative energy production across the globe.
4. Use graphing library to generate a heat map from the resulting data.

Need an account?

support@plantpredict.com

PLANT PREDICT

POWERED BY

First Solar.

First Solar®