Standardization of data ontologies and research data management

Michael Götte
Research Data Steward
Helmholtz Zentrum Berlin

The principles of Linked Data

(slide from https://jakub.klímek.com/nswi144)

1) Use URIs as names for things.

2) Use HTTP URIs (URLs) so that people can look up those names.

3) When someone looks up a URI/URL, provide useful information.

4) Include links to other URIs/URLs so that they can discover more things.

Linked Open Data (LOD)

- No standard file formats
- Data can be stored in files, databases, web pages, needs to be „linkable“
- Context is provided through links to other entities (e.g. Ontology class definitions)

See: https://jakub.klímek.com/nswi144 for some lecture notes

Exploring semantics

Published (alpha) thin film solar cell ontology:
https://matportal.org/ontologies/TFSCO/?p=summary

We use this ontology to connect Nomad entities with the semantic:

```python
class SpinCoating(WetChemicalDeposition):
    '''Base class for spin coating of a sample'''
    m_def = Section(
        #Link to ontology class 'spin coating'
        links = ['http://purl.obolibrary.org/obo/CHMO_0001472']
    )
```

```python
slot_die_head_distance_to_thinfilm = Quantity(
    # Link to ontology class 'slot die head distance to thinfilm' and 'slot die head distane to thinfilm setting datum'
    links=['http://www.semanticweb.org/ot2661/ontologies/2022/8/TFSCO#TFSCO_00005034',
           'http://www.semanticweb.org/ot2661/ontologies/2022/8/TFSCO#TFSCO_00005044'],
    type=np.dtype(
        np.float64),
    unit=('mm'),
    a_eln=dict(
        component='NumberEditQuantity',
        defaultDisplayUnit='mm',
        props=dict(
            minValue=0)))
```

http://purl.obolibrary.org/obo/CHMO_0001472

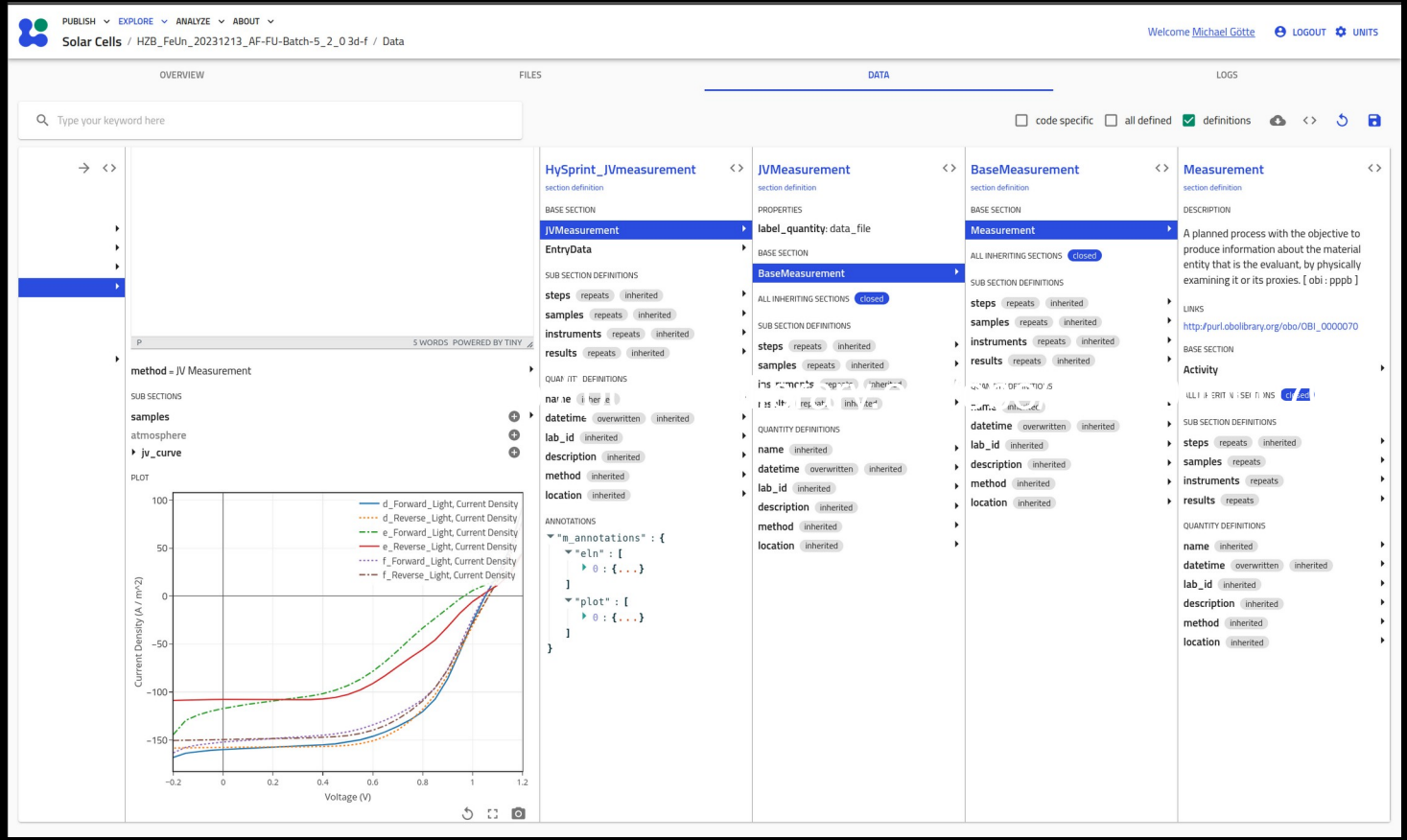
https://purl.archive.org/tfsco/TFSCO_00005034

Showcase in Nomad on Data annotations

https://nomad-hzb-se.de/nomad-oasis/gui/user/uploads/upload/id/JQr0kd9SQmevZ_BF_JGlcg/entry/id/XLMy_MNIe_hJrfOZ_zWbOyB_6cvt/data/results/properties/optoelectronic/solar_cell/efficiency

Go to Nomad (https://nomad-hzb-se.de/nomad-oasis/gui/about/information)

Showcase in Nomad on Data annotations (screen shots)



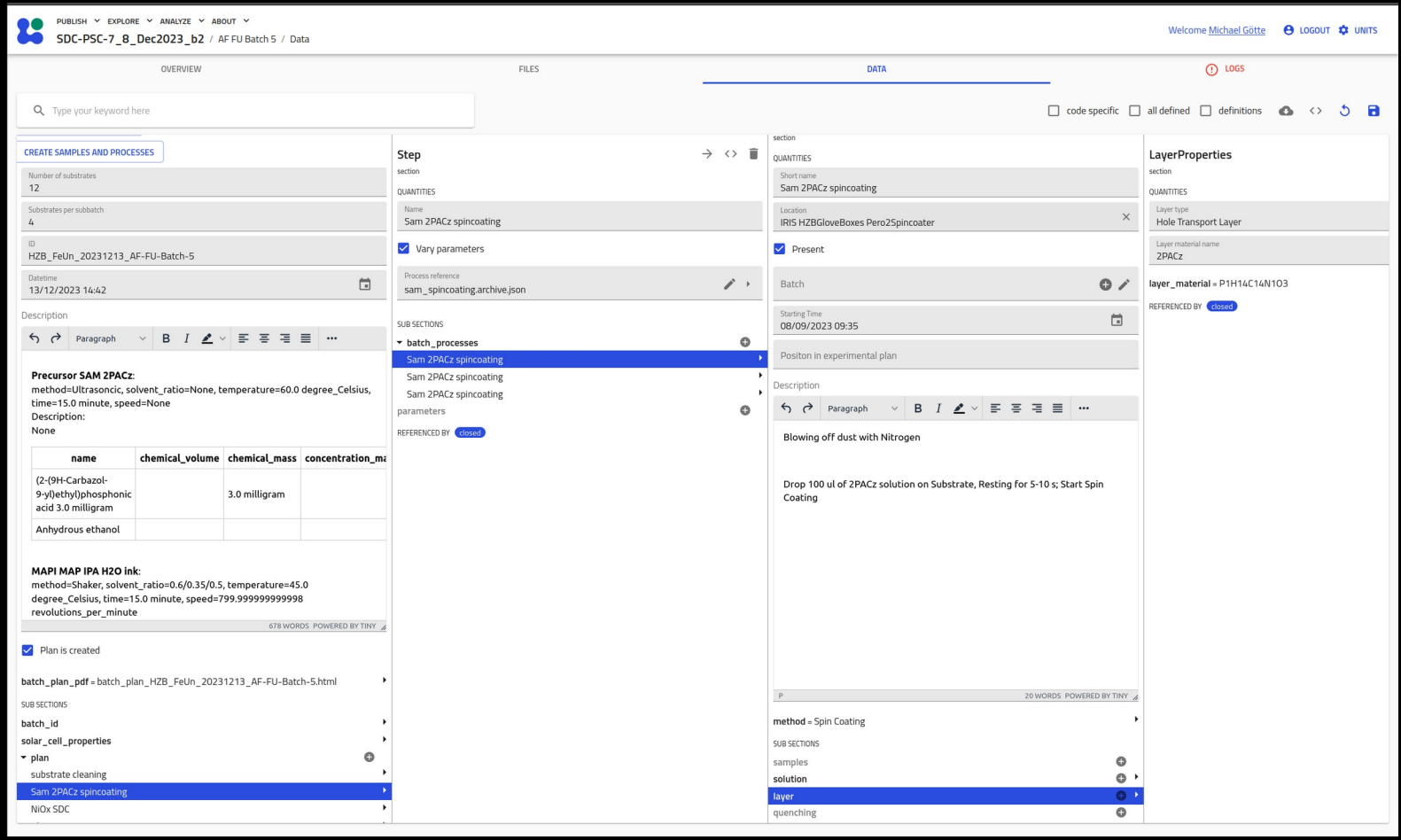
http://purl.obolibrary.org/obo/OBI_0000070

Showcase in Nomad on experimental plan

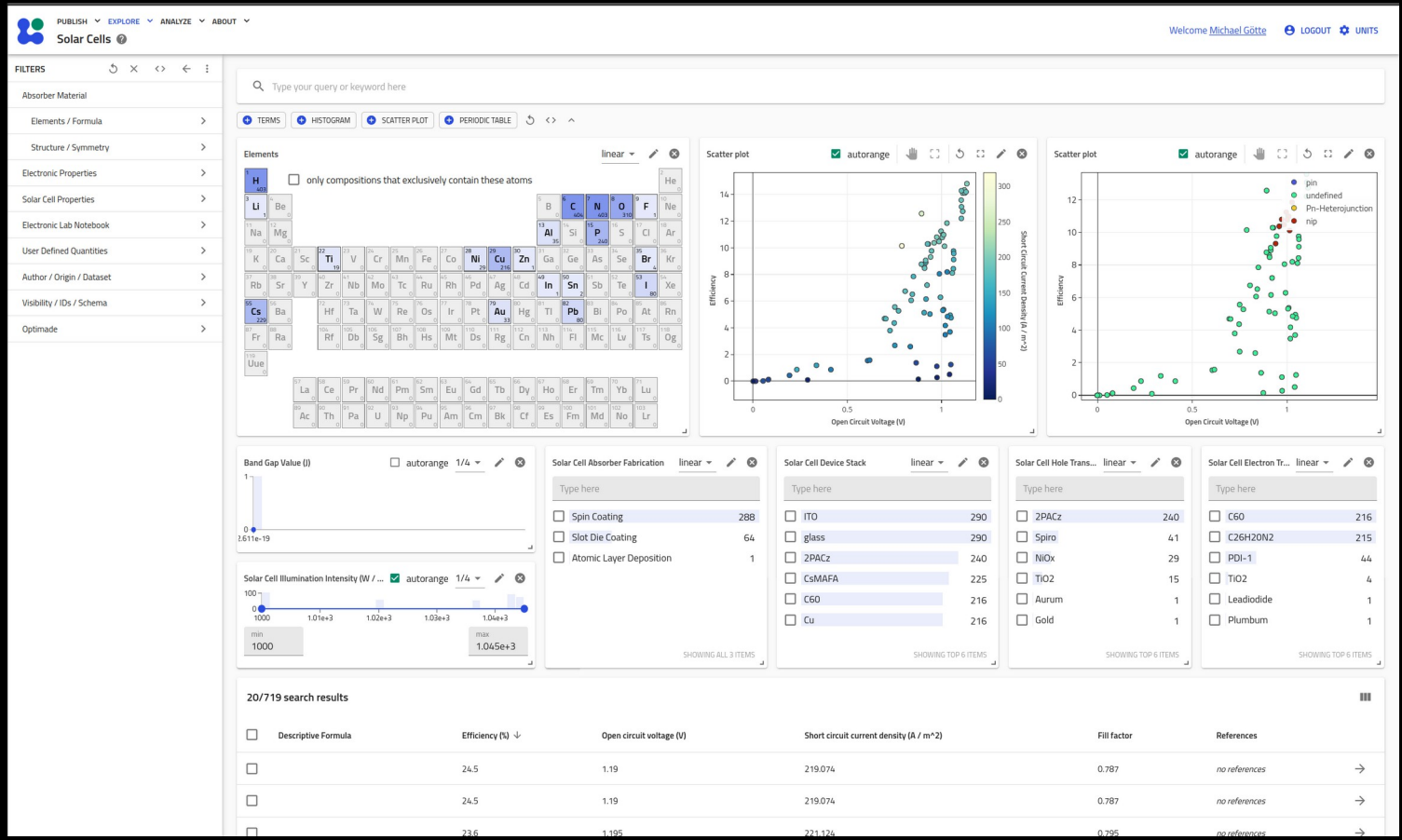
Go to Nomad (
https://nomad-hzb-se.de/nomad-oasis/gui/search/solarcells?upload_id=JQr0kd9SQmevZ_BF_JGlcg
)

https://nomad-hzb-se.de/nomad-oasis/gui/user/uploads/upload/id/JQr0kd9SQmevZ_BF_JGlcg/entry/id/iW1BoRMcxzWgEn-VvO4Gjg8aSlbj

Showcase in Nomad on experimental plan



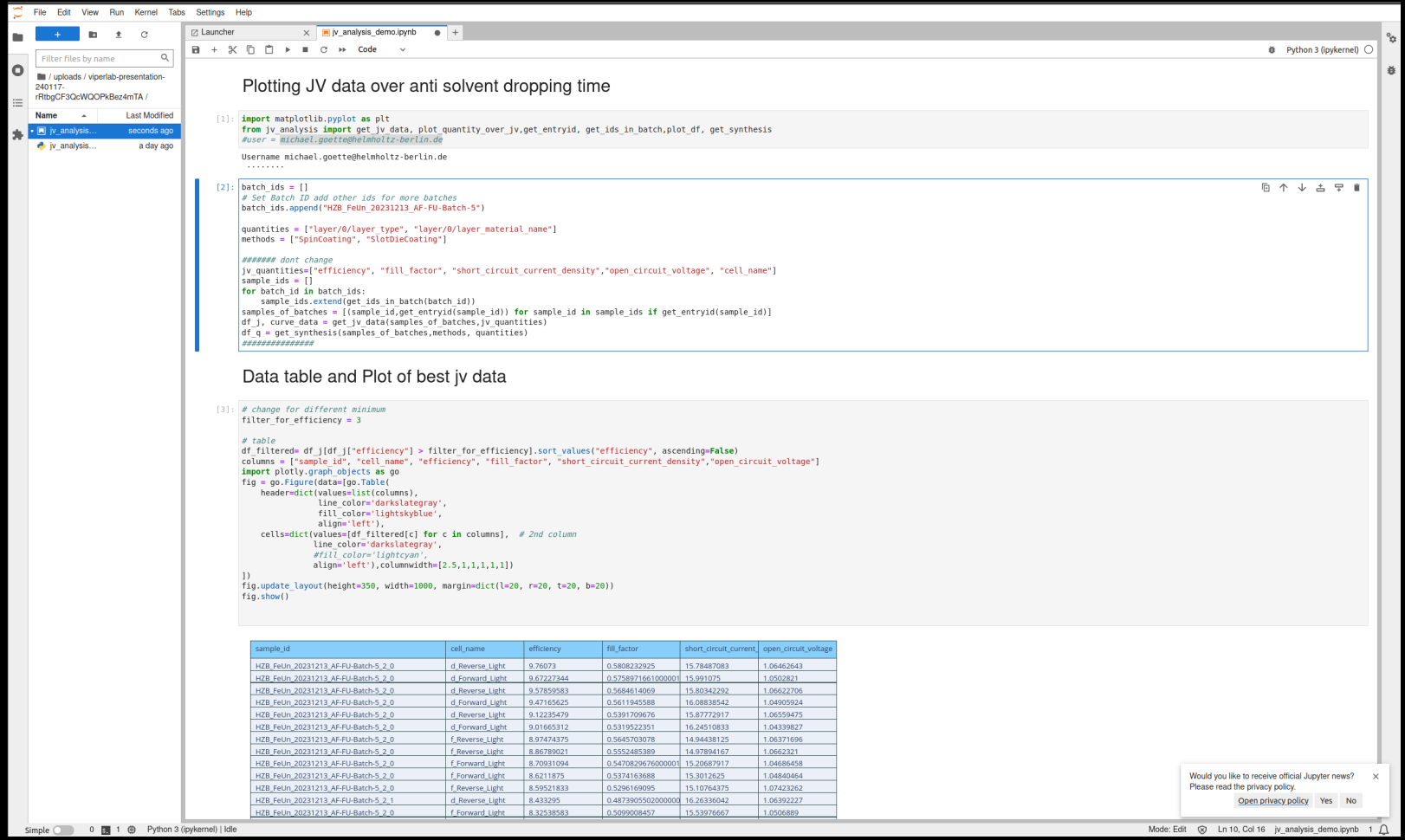
Showcase in Nomad on experimental plan



Showcase in Nomad on using jupyter hub

Go to Nomad (
https://nomad-hzb-se.de/nomad-oasis/north/user/michaelgoette/jupyter/lab/tree/uploads/data-analysis-feray-I3tq0kSiQDWS5Uxj91m7rw/jv_analysis_feray.ipynb
)

Showcase in Nomad on using jupyter hub



Showcase in Nomad on using jupyter hub

File Edit View Run Kernel Tabs Settings Help

Launcher | jv_analysis_demo.ipynb

Code | Python 3 (ipykernel)

Filter files by name

/ uploads / viperlab-presentation-240117-rRtbgCF3QcWQOPkBez4mTA /

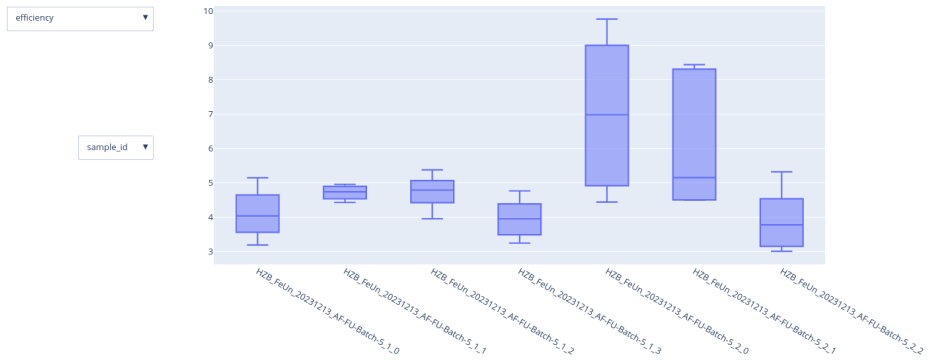
Name	Last Modified
jv_analysis...	a minute ago
jv_analysis...	a day ago

[4]:
```
# building the layer stack
####### dont change
import pandas as pd
def combine_columns(row):
    layers =[row[c]  for c in row.index if "layer_material_name" in c and not pd.isnull(row[c])]
    return "_".join(layers)
df_q['layer_stack'] =df_q.apply(combine_columns, axis=1)
####### dont change
```

Box plot for different layer stacks

[5]:
```
#### change filter for minimum efficiency
filter_for_efficiency = 3

########## dont change
df = df_q.merge(df_j,on=["entry_id","sample_id"])
plot_df(df[df["efficiency"] > filter_for_efficiency], jv_quantities,q_quantities=["layer_stack","sample_id", "jv_name"])
########## dont change
```



JV Curves of the best measurements by layer stack

[10]:
```
# finding best curves
idx_max = df.reset_index().groupby(['layer_stack'])["efficiency"].idxmax()

# plotting
plt.figure(figsize=(14,8))
for index, row in df.iloc[idx_max].iterrows():
```

Would you like to receive official Jupyter news? Please read the privacy policy.
Open privacy policy | Yes | No

Simple | Python 3 (ipykernel) | Idle | Mode: Command | Ln 1, Col 1 | jv_analysis_demo.ipynb

Showcase in Nomad on using jupyter hub

File Edit View Run Kernel Tabs Settings Help

Launcher | jv_analysis_demo.ipynb

Code | Python 3 (ipykernel)

Filter files by name

/ uploads / viperlab-presentation-240117-rRtbgCF3QcWQOPkBez4mTA /

Name	Last Modified
jv_analysis…	a minute ago
jv_analysis…	a day ago

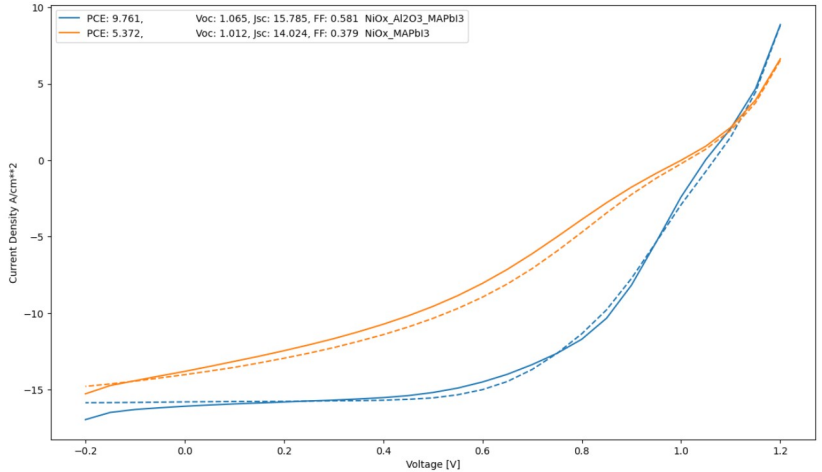
JV Curves of the best measurements by layer stack

[6]:
```
# finding best curves
idx_max = df.reset_index().groupby(['layer_stack'])["efficiency"].idxmax()

# plotting
plt.figure(figsize=(14,8))
for index, row in df.iloc[idx_max].iterrows():
    if row["efficiency"] < filter_for_efficiency:
        continue
    cell = row['cell_name'][0]

    curve_fwd = curve_data.get(f"{index}_{cell}_Forward_Light")
    curve_rev = curve_data.get(f"{index}_{cell}_Reverse_Light")

    p = plt.plot(curve_fwd.get("voltage"), curve_fwd.get("current_density"), label=f"PCE: {round(row['efficiency'],3)},\
            Voc: {round(row['open_circuit_voltage'],3)}, Jsc: {round(row['short_circuit_current_density'],3)}, FF: {round(row['fill_factor'],3)}  {row['layer_stack']}")
    plt.plot(curve_rev.get("voltage"), curve_rev.get("current_density"), "--",color=p[0].get_color())
plt.xlabel("Voltage [V]")
plt.ylabel("Current Density A/cm**2")
plt.legend()
plt.show()
```



[ ]:

[ ]:

Would you like to receive official Jupyter news? Please read the privacy policy.
Open privacy policy | Yes | No

Simple | Python 3 (ipykernel) | Idle | Mode: Command | Ln 1, Col 1 | jv_analysis_demo.ipynb