



23,000 annual simulation users



nanoHUB: getting started guide to tool developers

Develop and publish tools in nanoHUB

Make your research reproducible and your workflows and data FAIR

Tanya Faltens, Daniel Mejia, Steven Clark, Juan Carlos Verduzco & Ale Strachan*

* strachan@purdue.edu

School of Materials Engineering &

Purdue University

West Lafayette, Indiana USA

Overview

1. Why publish tools & apps in nanoHUB?
 - Tools are publications (DOIs and indexed by Web of Science)
 - Share your work with your community (22,000+ annual sim users)
2. Various tool and app types
 - Apps, workflows, Jupyter notebooks, commercial codes, X11 GUIs
3. Sim2Ls, FAIR workflows and data
 - Develop and publish Sim2Ls
4. Developing Apps
 - Connecting Sim2Ls to Jupyter and Web Apps
5. Tool Publication process
 - Register, deploy, test, and publish
6. Development environment
 - A Unix development environment (Jupyter or Linux desktop)
7. Simulation and data as a service
 - Launching tools and querying the ResultsDB



6. Accessing nanoHUB apps, tools, & data

Science codes, data tools & repos

Schred

siesta

NEMO5

Polymer Modeler

Abinit

Padre

NAMD Scalable Mo

TensorFlow

Keras

OpenKIM

CITRINE INFORMATICS

The Materials Project

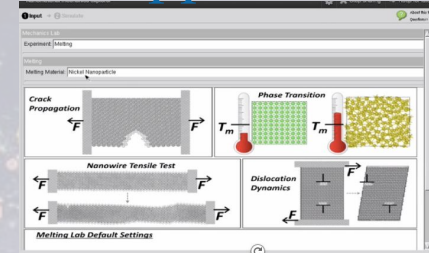
WolframAlpha computational intelligence.

nanoHUB services

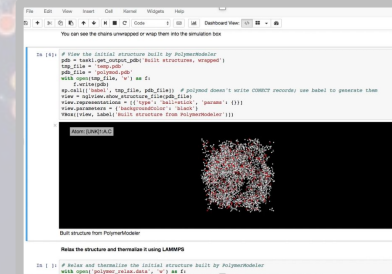
Rappture
Jupyter
Sim2Ls
Simulation cache
ResultsDB



Web Apps & Tools

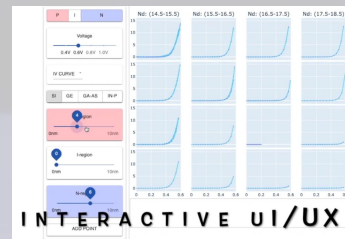
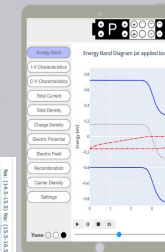


Science workflows



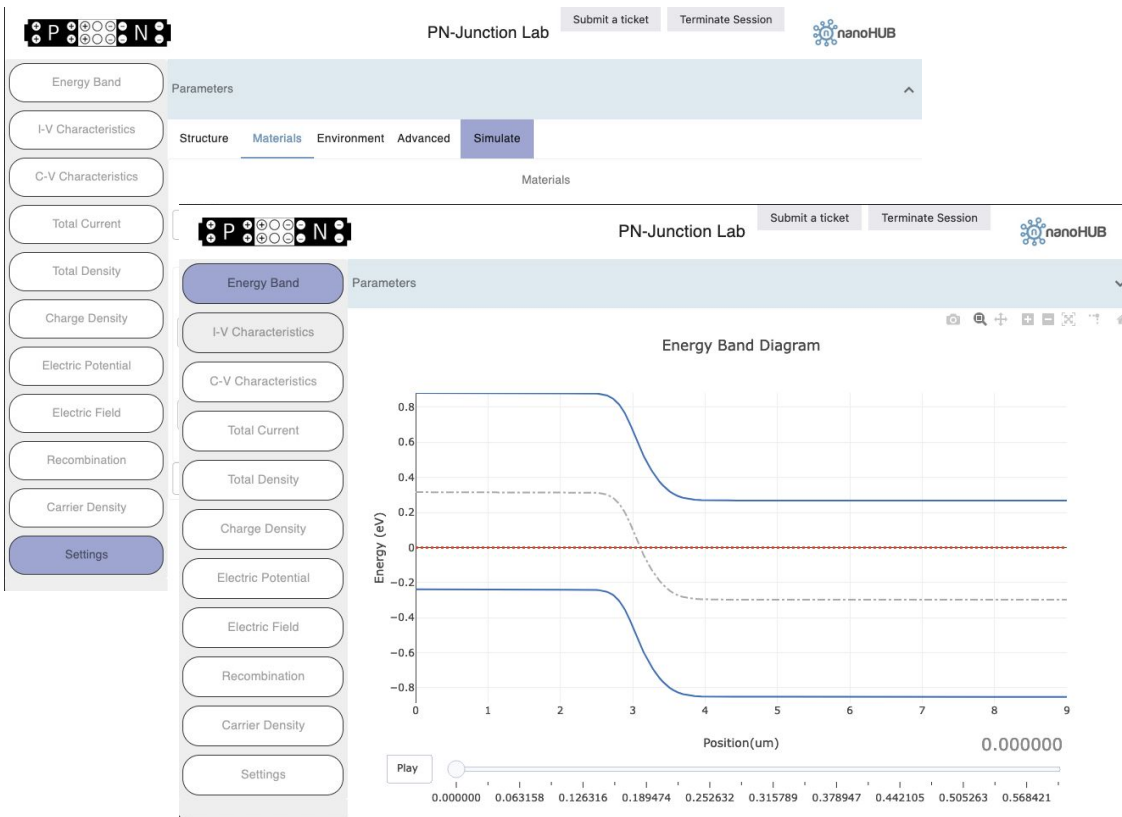
Web Service

Desktop, mobile, & embedded apps

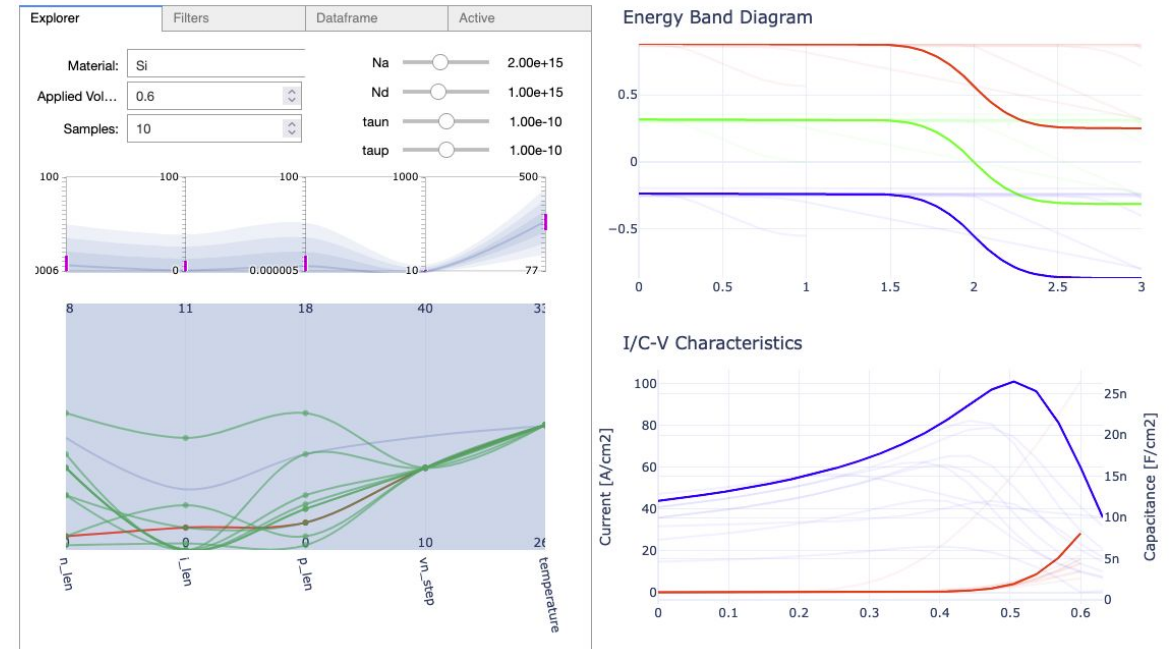


Data exploration tools

Web Services: overview

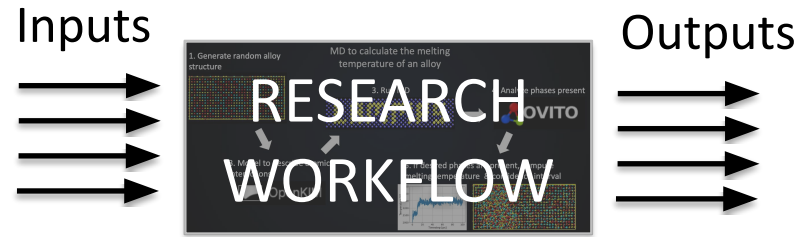


<https://nanohub.org/tools/pnjunctionapp>

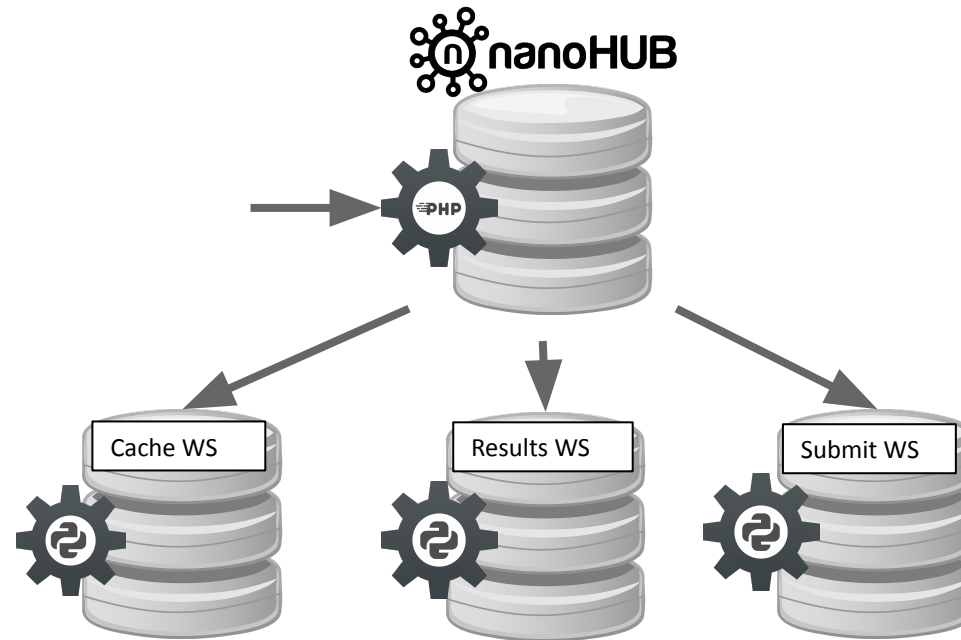
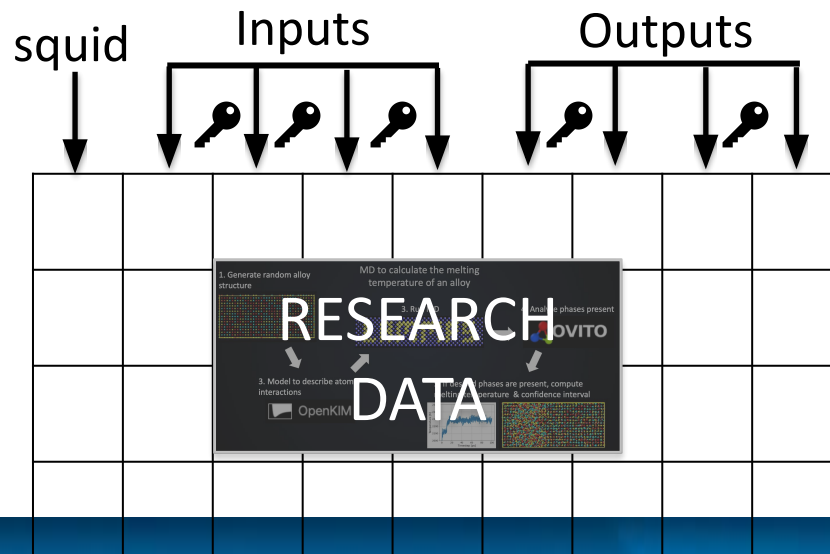


<https://nanohub.org/tools/nhremote> (vizexplorer)

Web Services: overview



- Sim2L/Rappture cached and indexed
 - Sim2L/Rappture schemas introspection
 - Results are indexed (research data)
 - REST API to query research data
 - REST API to submit Sim2L/Rappture



- Python wrapper to access REST API

<https://nanohub.org/api/dbexplorer/dbexplorer/>
<https://nanohub.org/developer/api/endpoint/dbexplorer>



<https://pypi.org/project/nanohub-remote/>

Web Services: access ResultsDB

1. Create a Session

Web Services Session

Using the authentication data nanohub-remote creates a Generic session (nr.Tools(auth_data))

```
In [4]: 1 session = nr.Session(auth_data)
```

To view all available tools on the results database use the tools end point

```
In [5]: 1 req_json = session.requestPost('dbexplorer/dbexplorer/tools?simtool=true')
2 req_json = req_json.json()
3 pd.DataFrame([p for p in req_json["results"]])
```

Out[5]:

| | last_version | tool_id | total_versions |
|---|--------------|-----------------|----------------|
| 0 | r9 | introtosimtools | 2 |
| 1 | r13 | meltrocas | 1 |
| 2 | r34 | meltingkim | 3 |
| 3 | r21 | mdsandbox | 1 |
| 4 | r39 | meltneas | 3 |
| 5 | r9 | st4pnjunction | 2 |

Query tool tool inputs and output

Given a specific tool is possible to query their schema inputs

```
In [6]: 1 SIM2L = "st4pnjunction"
2 req_json = session.requestPost('dbexplorer/dbexplorer/search', data={'tool': SIM2L})
3 req_json = req_json.json()
```

2. Query REST API Endpoints

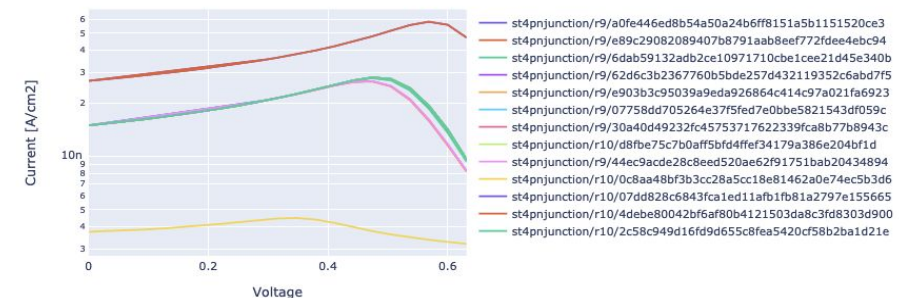
```
In [13]: 1 N_LEN = results[0]["input.n_len"]
2 I_LEN = results[0]["input.i_len"]
3 VOLTAGE = [0,0.6]
4 TEMPERATURE = [300,300]
5 MATERIALP = "Si"
6 IMPURITY = "false"
7
8 search = {
9     'tool': SIM2L,
10    'simtool': True,
11    'filters': json.dumps([
12        {'field': 'input.n_len', 'value': N_LEN, 'operation': '='},
13        {'field': 'input.temperature', 'value': TEMPERATURE[0], 'operation': '>='},
14        {'field': 'input.temperature', 'value': TEMPERATURE[1], 'operation': '<='},
15        {'field': 'input.i_len', 'value': I_LEN, 'operation': '='},
16        {'field': 'input.materialp', 'value': MATERIALP, 'operation': '='},
17        {'field': 'input.impurity', 'value': IMPURITY, 'operation': '='},
18        {'field': 'input.voltage', 'value': 0, 'operation': '='},
19    ]),
20    'results': json.dumps([
21        'input.Na',
22        'input.Nd',
23        'input.n_len',
24        'input.i_len',
25        'input.p_len',
26        'output.Parameters',
27        'output.IV Characteristic',
28    ])
29 ]}
30 req_json = session.requestPost('dbexplorer/dbexplorer/search', data=search)
31 req_json = req_json.json()
32 pd.DataFrame(req_json['results'])
```

3. Get results as JSON

4. Explore results

```
In [14]: 1 layout1 = {
2     'title': "IV Characteristics",
3     'xaxis': {
4         'title': 'Voltage',
5     },
6     'yaxis': {
7         'title': 'Current [A/cm2]',
8         'type': 'log',
9     },
10 }
11 traces = []
12 for res in req_json['results']:
13     traces.append({
14         'type': 'scatter',
15         'name': res['squid'],
16         'x': res['output.IV Characteristic']['voltage'],
17         'y': res['output.IV Characteristic']['function'],
18         'text': "N: " + str(res['input.n_len']) + ", P: " + str(res['input.p_len']) + "<br> " + "Na: " + str(res['input.na']) + ", P: " + str(res['input.p_len'])
19     })
20 fig = FigureWidget(traces, layout1)
21 fig
```

IV Characteristics



See details and implementation at:
<https://nanohub.org/tools/nhremote>

Web Services: access ResultsDB

Auth

Allows users to access nanoHUB resources from outside nanoHUB by creating authentication tokens, **These services are not required if Python/Jupyter are executed** on nanoHUB

- [nanoHUB web API Documentation](#)
- [register account to enable remote access](#)
- [types of Authentication](#)
- [Deprecated library](#)



Tools

Allows users to access results and run new simulations

- [Exploring available nanoHUB tools](#)
- [Exploring available nanoHUB simtools](#)
- [nanohubtools Library](#)
- [nanohubtools Library \(App Mode\)](#)



Results Databases

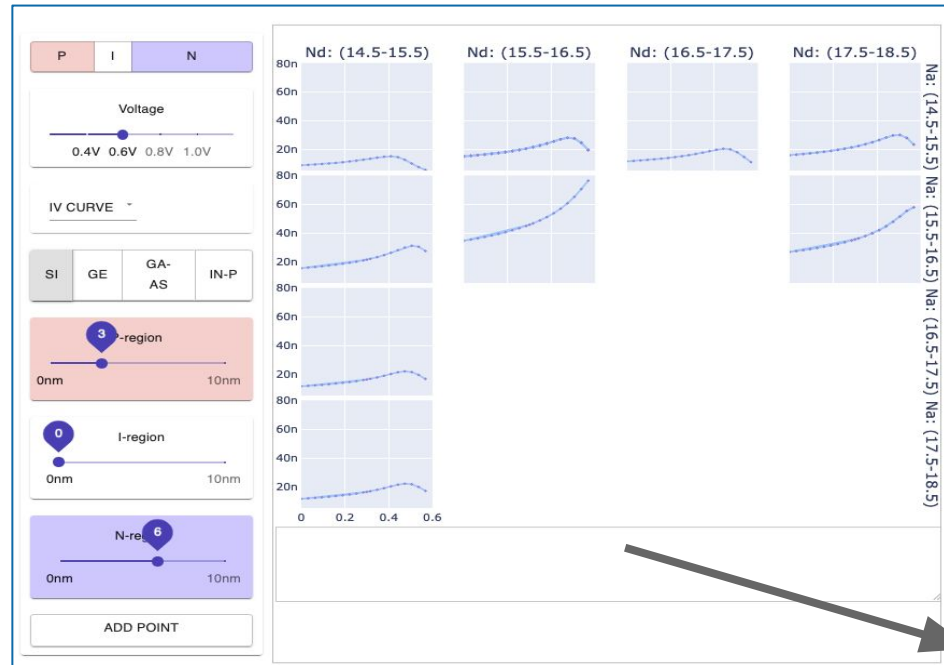
Allows users to query outputs from previous results

- [Querying results rappture](#)
- [Querying results simtool](#)
- [Exploring Exploration using visualizations](#)
- [Simple Machine learning models](#)

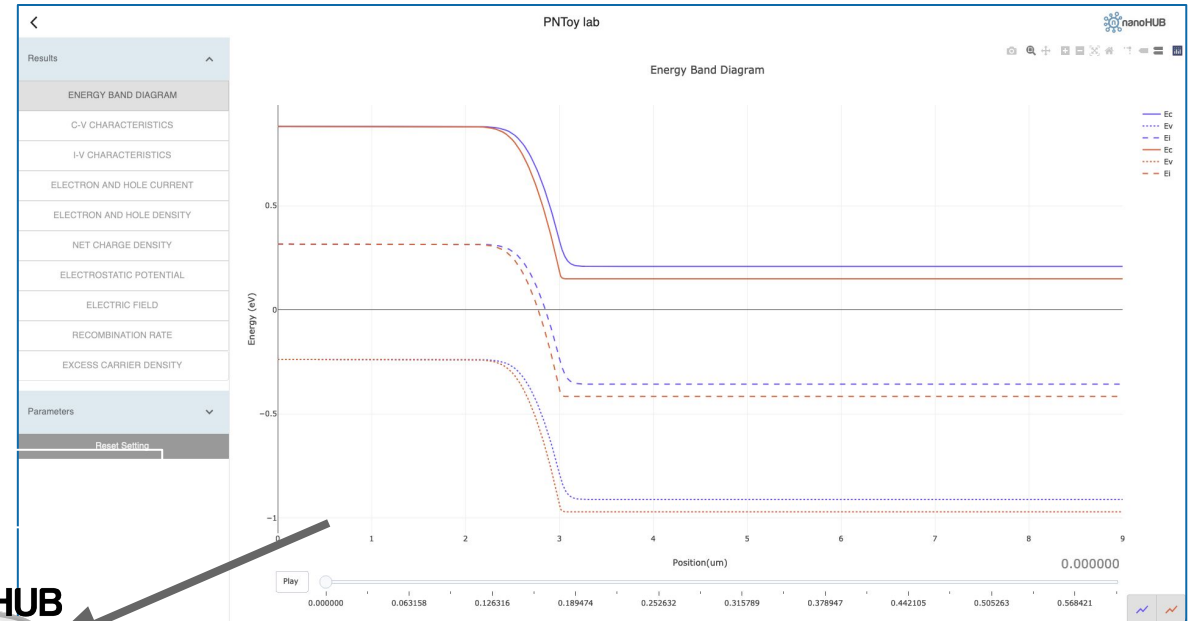


See details and implementation at:
<https://nanohub.org/tools/nhremote>

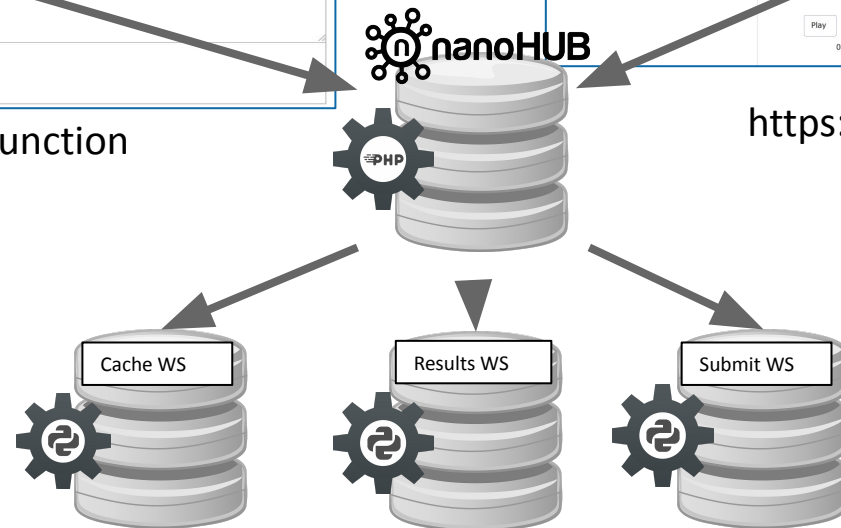
Web Apps: exploring results and submitting jobs



<https://nanohub.org/tools/ex2pnjunction>

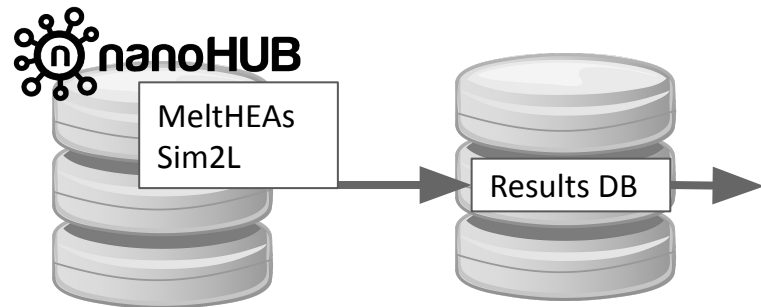
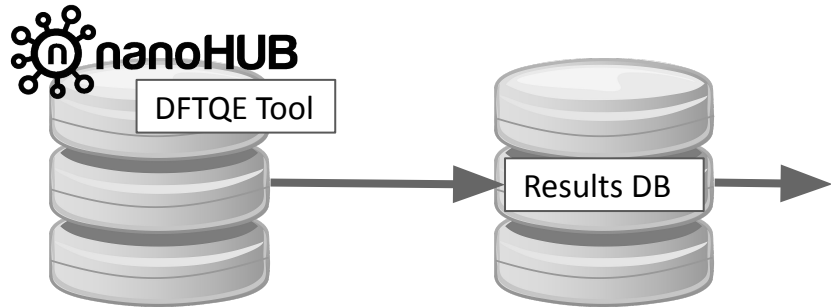


https://nanohub.org/js_apps/st4pnjunctionPROD.html



- JS or Jupyter Webapps based on REST API endpoints

Jupyter tools: exploring results and submitting jobs



<https://nanohub.org/tools/meltdashboard>

DFT Results Explorer Tool

Explore results database

Material: GaAs
 Functional: LDA
 Output: Bulk Modulus

Run new simulations

Grid Points: Min: 2, Max: 8, Points: 2
 Grid Cells: Min: 2, Max: 8, Points: 2

Run Grid Points

1. Explore prior runs → 2. Find unexplored space → 3. Launch new simulations

Linear Space

GaAs – Bulk Modulus / GGA Functional

Linear Space

GaAs – Bulk Modulus / LDA Functional

Linear Space

GaAs – Bulk Modulus / LDA Functional

<https://nanohub.org/tools/dftexplorer>

