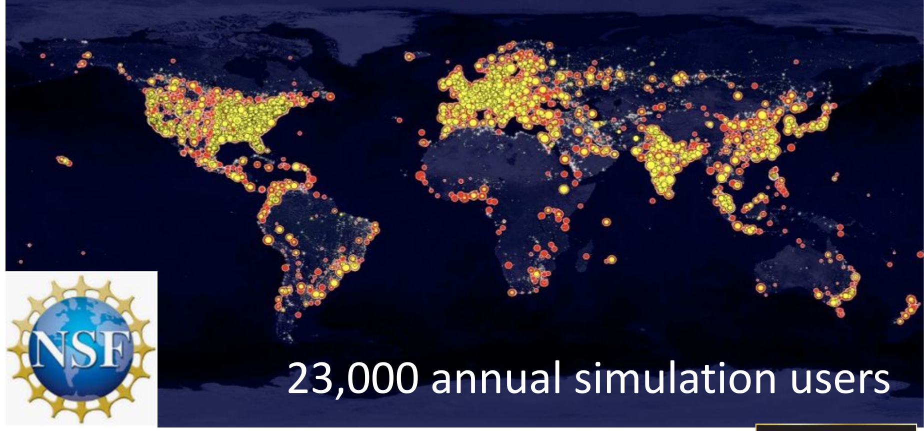


nanoHUB: getting started guide for tool developers

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

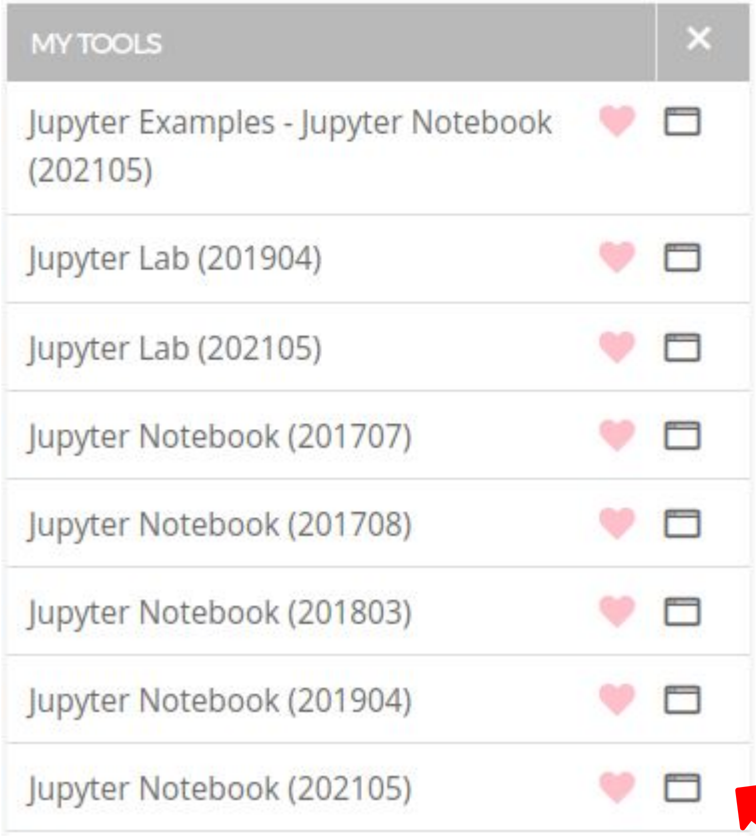
Overview

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

Development environments

Two main development environments

Jupyter


MY TOOLS

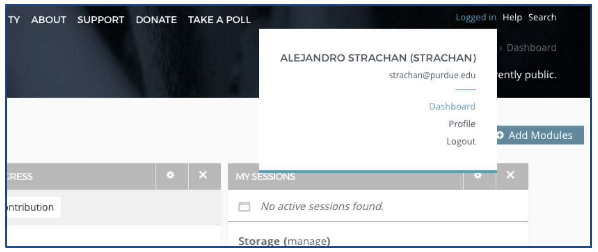
Jupyter Examples - Jupyter Notebook (202105)	
Jupyter Lab (201904)	
Jupyter Lab (202105)	
Jupyter Notebook (201707)	
Jupyter Notebook (201708)	
Jupyter Notebook (201803)	
Jupyter Notebook (201904)	
Jupyter Notebook (202105)	

Launch tool

https://nanohub.org/tools/jupyter70

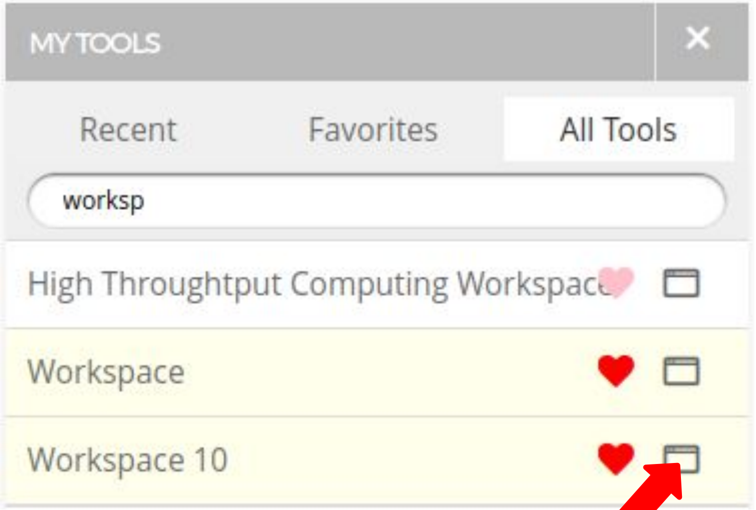
Remember to check for the latest Jupyter tool

1. Login to nanoHUB
2. Go to your dashboard



3. Select your tool

Linux workspace


MY TOOLS

Recent | Favorites | All Tools

worksp

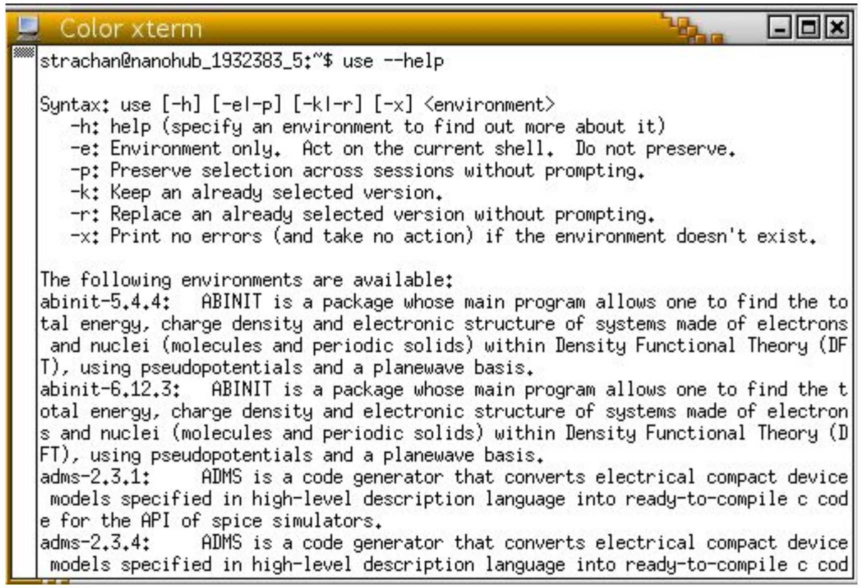
High Throughtput Computing Workspace	
Workspace	
Workspace 10	

Launch tool

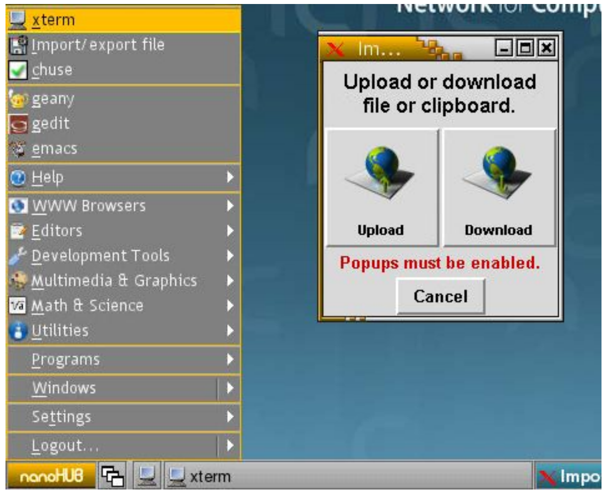
https://nanohub.org/tools/workspace10

Tool development environment: workspace

- Docker image with Debian OS
- Persistent user owned storage
- Persistent tool sessions accessed with web browser
- Scientific software packages accessible with the *use* command
  - Molecular dynamics
  - Numerical solvers
  - Workflow management
  - Visualization
  - Quantum Chemistry
  - … and many more



```
strachan@nanohub_1932383_5:~$ use --help

Syntax: use [-h] [-e|-p] [-k|-r] [-x] <environment>
   -h: help (specify an environment to find out more about it)
   -e: Environment only.  Act on the current shell.  Do not preserve.
   -p: Preserve selection across sessions without prompting.
   -k: Keep an already selected version.
   -r: Replace an already selected version without prompting.
   -x: Print no errors (and take no action) if the environment doesn't exist.

The following environments are available:
abinit-5.4.4:    ABINIT is a package whose main program allows one to find the to
tal energy, charge density and electronic structure of systems made of electrons
 and nuclei (molecules and periodic solids) within Density Functional Theory (DF
T), using pseudopotentials and a planewave basis.
abinit-6.12.3:   ABINIT is a package whose main program allows one to find the t
otal energy, charge density and electronic structure of systems made of electron
s and nuclei (molecules and periodic solids) within Density Functional Theory (D
FT), using pseudopotentials and a planewave basis.
adms-2.3.1:      ADMS is a code generator that converts electrical compact device
 models specified in high-level description language into ready-to-compile c cod
e for the API of spice simulators.
adms-2.3.4:      ADMS is a code generator that converts electrical compact device
 models specified in high-level description language into ready-to-compile c cod
```

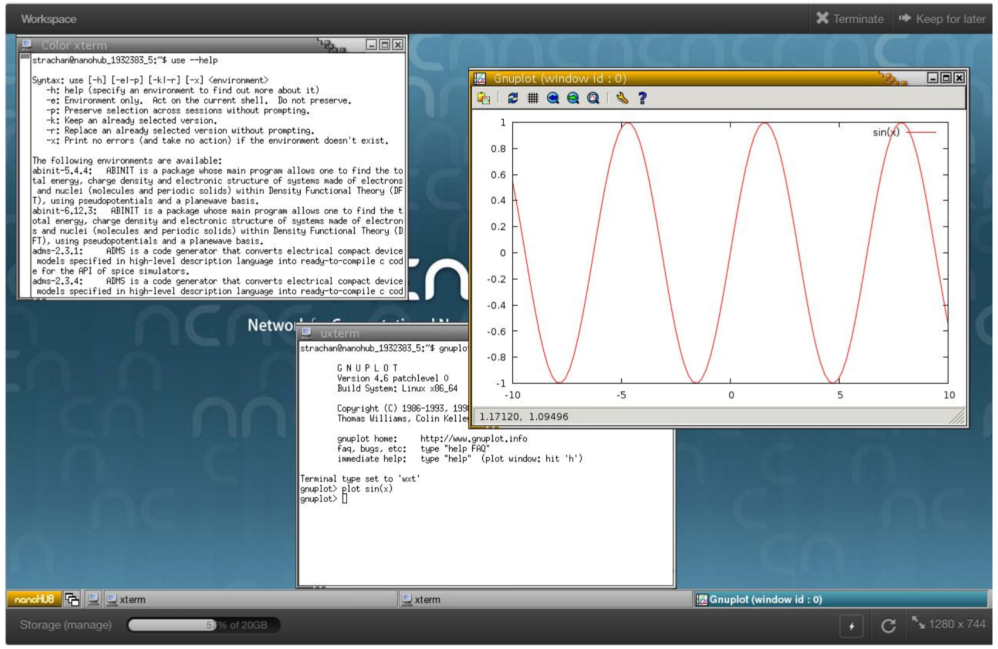


Upload or download file or clipboard.

Popups must be enabled.

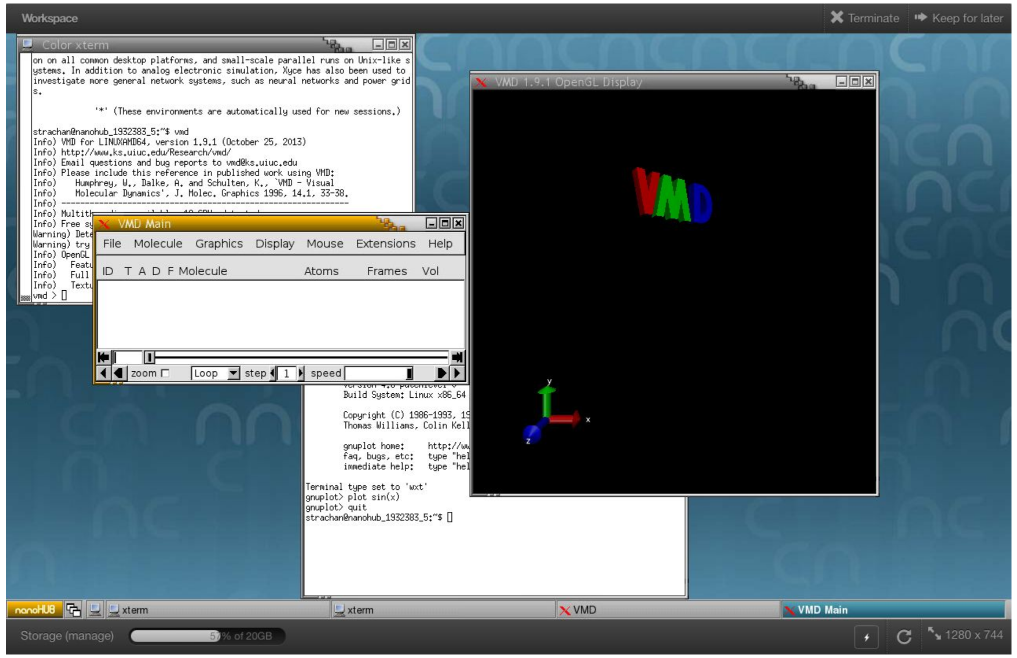
- HPC/HTC resource access with the submit command
  - Purdue clusters
    - OpenMP and MPI
    - Computational GPU
    - Bare metal or containers
  - Open Science Grid
    - Parametric sweeps
- File transfer service

Tool development environment: workspace



```
strachan@nanohub_1932383_5:~$ use --help

Syntax: use [-h] [-e|-p] [-k|-r] [-x] <environment>
   -h: help (specify an environment to find out more about it)
   -e: Environment only.  Act on the current shell.  Do not preserve.
   -p: Preserve selection across sessions without prompting.
   -k: Keep an already selected version.
   -r: Replace an already selected version without prompting.
   -x: Print no errors (and take no action) if the environment doesn't exist.

The following environments are available:
abinit-5.4.4:   ABINIT is a package whose main program allows one to find the to
tal energy, charge density and electronic structure of systems made of electrons
 and nuclei (molecules and periodic solids) within Density Functional Theory (DF
T), using pseudopotentials and a planewave basis.
abinit-6.12.3:   ABINIT is a package whose main program allows one to find the t
otal energy, charge density and electronic structure of systems made of electron
s and nuclei (molecules and periodic solids) within Density Functional Theory (D
FT), using pseudopotentials and a planewave basis.
adms-2.3.1:    ADMS is a code generator that converts electrical compact device
 models specified in high-level description language into ready-to-compile c cod
e for the API of spice simulators.
adms-2.3.4:    ADMS is a code generator that converts electrical compact device
 models specified in high-level description language into ready-to-compile c cod
```

```
strachan@nanohub_1932383_5:~$ gnuplo

        G N U P L O T
        Version 4.6 patchlevel 0
        Build System: Linux x86_64

        Copyright (C) 1986-1993, 199
        Thomas Williams, Colin Kelle

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type set to 'wxt'
gnuplot> plot sin(x)
gnuplot> 
```



```
on on all common desktop platforms, and small-scale parallel runs on Unix-like s
ystems. In addition to analog electronic simulation, Xyce has also been used to
investigate more general network systems, such as neural networks and power grid
s.

            '*' (These environments are automatically used for new sessions.)

strachan@nanohub_1932383_5:~$ vmd
Info) VMD for LINUXAMD64, version 1.9.1 (October 25, 2013)
Info) http://www.ks.uiuc.edu/Research/vmd/
Info) Email questions and bug reports to vmd@ks.uiuc.edu
Info) Please include this reference in published work using VMD:
Info)    Humphrey, W., Dalke, A. and Schulten, K., `VMD - Visual
Info)    Molecular Dynamics', J. Molec. Graphics 1996, 14.1, 33-38.
Info) -------------------------------------------------------------
Info) Multith
Info) Free sy
Warning) Dete
Warning) try
Info) OpenGL
Info)   Featu
Info)   Full
Info)   Textu
vmd > 
```

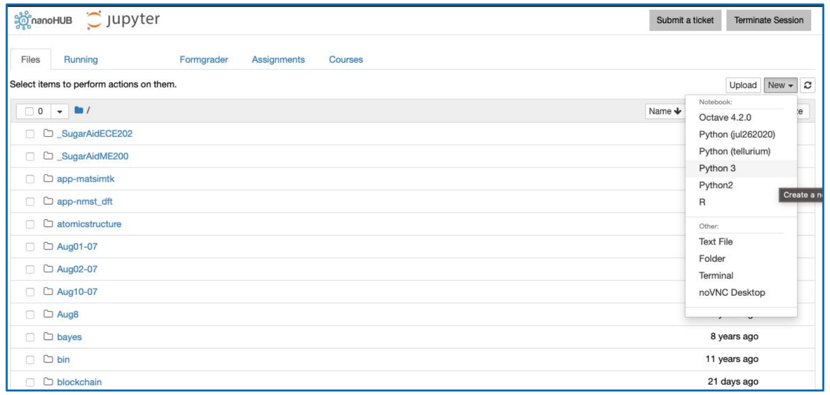
```
        Build System: Linux x86_64

        Copyright (C) 1986-1993, 19
        Thomas Williams, Colin Kell

        gnuplot home:     http://ww
        faq, bugs, etc:   type "hel
        immediate help:   type "hel

Terminal type set to 'wxt'
gnuplot> plot sin(x)
gnuplot> quit
strachan@nanohub_1932383_5:~$ 
```

- Editors (text and UI interfaces)
- make
- UI building
  - Rappture
  - Qt, PyQt
  - MATLAB

- Scripting languages
  - Python2/3
  - Octave
  - MATLAB
  - tcl/tk
  - ruby

- Compiled languages
  - C/C++
  - Fortran
  - Java

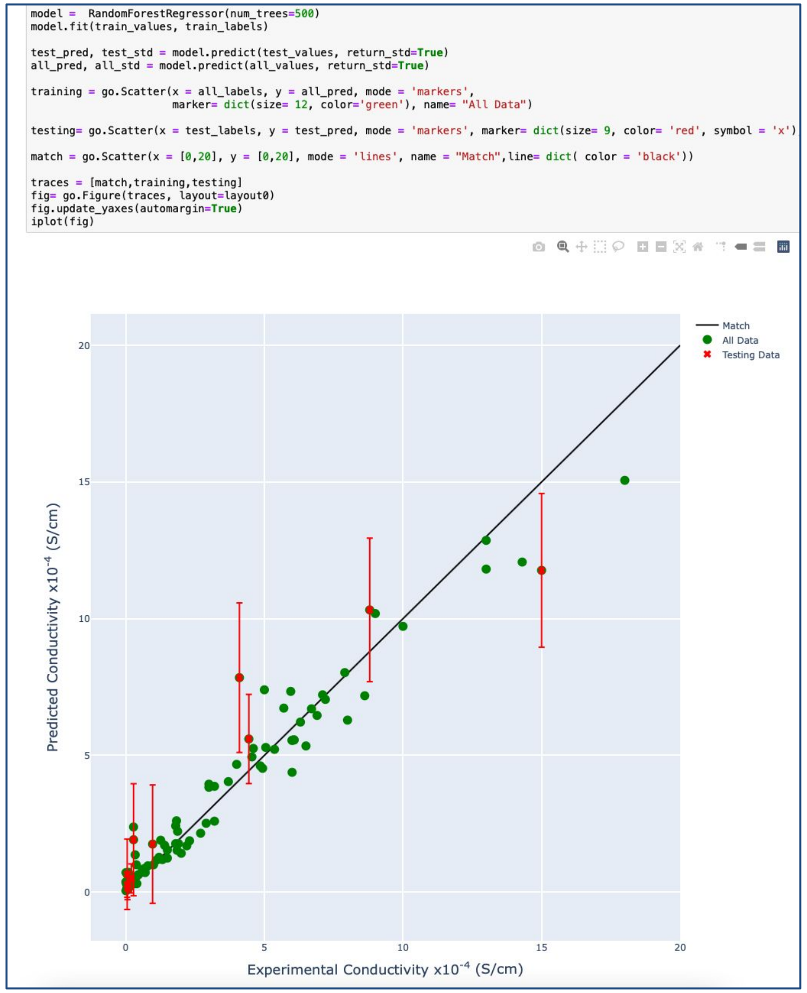
Tool development environment: Jupyter



Select items to perform actions on them.

```
_SugarAidECE202
_SugarAidME200
app-matsimtk
app-nmst_dft
atomicstructure
Aug01-07
Aug02-07
Aug10-07
Aug8
bayes                8 years ago
bin                 11 years ago
blockchain           21 days ago
```

- Kernel support for several languages
  - Python3
  - MATLAB
  - Octave
  - R
  - Custom tailored environments
- Software sources
  - conda
  - pip
- HUBzero software
  - hublib
  - Sim2Ls



```
model =  RandomForestRegressor(num_trees=500)
model.fit(train_values, train_labels)

test_pred, test_std = model.predict(test_values, return_std=True)
all_pred, all_std = model.predict(all_values, return_std=True)

training = go.Scatter(x = all_labels, y = all_pred, mode = 'markers',
                      marker= dict(size= 12, color='green'), name= "All Data")

testing= go.Scatter(x = test_labels, y = test_pred, mode = 'markers', marker= dict(size= 9, color= 'red', symbol = 'x')

match = go.Scatter(x = [0,20], y = [0,20], mode = 'lines', name = "Match",line= dict( color = 'black'))

traces = [match,training,testing]
fig= go.Figure(traces, layout=layout0)
fig.update_yaxes(automargin=True)
iplot(fig)
```

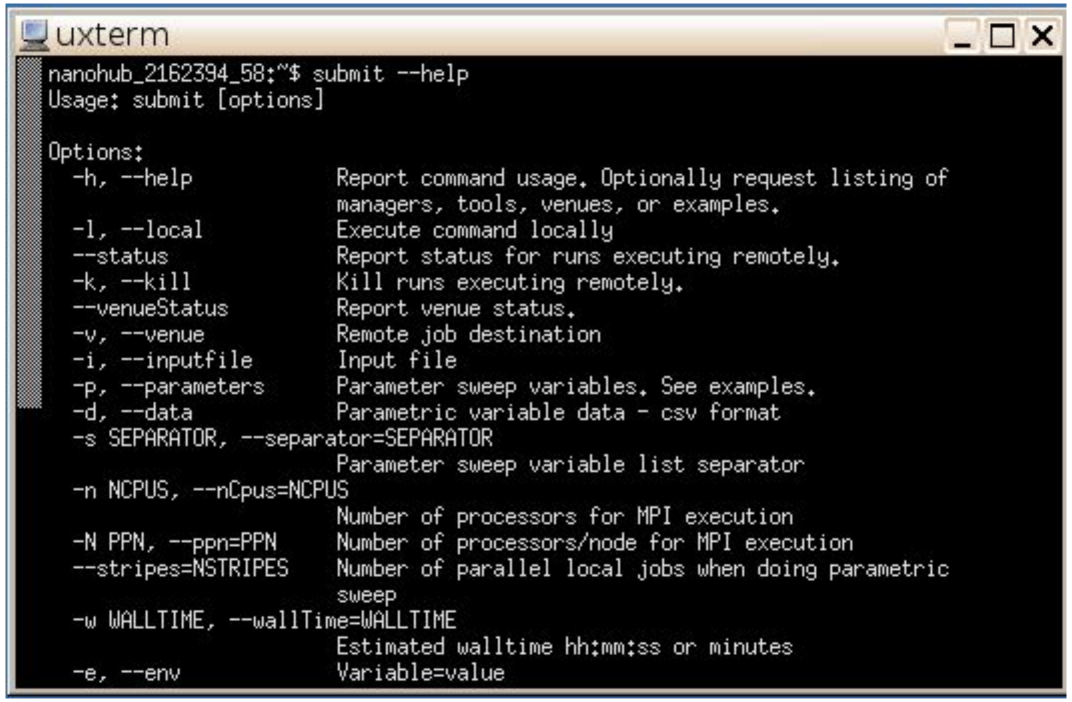
Match
All Data
Testing Data
Predicted Conductivity x10^{-4} (S/cm)
Experimental Conductivity x10^{-4} (S/cm)
0
5
10
15
20

Submitting intensive jobs to HPC/HTC resources

- High Performance Computing resources - for computational programs requiring multiple cores and/or long running times to complete.
- High Throughput Computing resources - for scenarios requiring many simulations with varying inputs to provide insight into the characteristics of a physical system under study.

submit command

- Available in Jupyter and Linux workspace environments
  - Command line
  - Jupyter notebooks
  - Sim2L integration
- More than 70 applications
  - Leading well recognized open source software packages serving multiple fields of study
  - Tool developer provided applications
- HPC/HTC access for tool developers and users alike



```
nanohub_2162394_58:~$ submit --help
Usage: submit [options]

Options:
  -h, --help            Report command usage. Optionally request listing of
                        managers, tools, venues, or examples.
  -l, --local           Execute command locally
  --status              Report status for runs executing remotely.
  -k, --kill            Kill runs executing remotely.
  --venueStatus         Report venue status.
  -v, --venue           Remote job destination
  -i, --inputfile       Input file
  -p, --parameters      Parameter sweep variables. See examples.
  -d, --data            Parametric variable data - csv format
  -s SEPARATOR, --separator=SEPARATOR
                        Parameter sweep variable list separator
  -n NCPUS, --nCpus=NCPUS
                        Number of processors for MPI execution
  -N PPN, --ppn=PPN     Number of processors/node for MPI execution
  --stripes=NSTRIPES    Number of parallel local jobs when doing parametric
                        sweep
  -w WALLTIME, --wallTime=WALLTIME
                        Estimated walltime hh:mm:ss or minutes
  -e, --env             Variable=value
```