A FAIR Approach to Data and Machine Learning Using funcX

Aristana Scourtas (aristana@uchicago.edu), Ben Blaiszik (blaiszik@uchicago.edu), Ryan Chard, ZhuoZhao Li, Logan Ward, Tyler Skluzacek, KJ Schmidt, Nathan Pruyne, Jim James, Ethan Truelove, Jonathon Gaff, Marcus Schwarting, Ian Foster (foster@anl.gov)

Dane Morgan, Ryan Jacobs, Paul Voyles, Michael Ferris, Jingrui Wei, Xiangguo Li

DLHub
Data and Learning Hub for Science
https://www.dlhub.org

https://www.dlhub.org
Data and ML are becoming key drivers of scientific progress
How do we use these models?

For a given study:

• Where is the code?
• Where are the trained models?
• Where are the training data?
• How can I reproduce these results?

Without all of these pieces, progress is drastically slowed

Need models and data to be FAIR:

Findable
Accessible
Interoperable
Reusable

Location of many ML models after a paper is finished
DLHub for FAIR Models

A simple way to find, share, publish, and run machine learning models

1. **ML Model Submission**
   - Collect ML models in common formats

2. **Container Creation**
   - Create portable containers of models and register in funcX

3. **Model Catalog**
   - Create a searchable index of data

---

**Model Publisher**
- DOI for citing model
- Landing page
- Run in the cloud
- API to model!

**Model Consumer**
- Find
- Run (in cloud)
• What if I want to evaluate a model from a paper?

```
structure_model = "npriyne_glbosid/cherukara_structure"
phase_model = "npriyne_glbosid/cherukara_phase"

# Load testing data
n_test = 10
intensity_threshold = 0.2
X = ft_test[0:n_test].tolist()

# Call to DLHub to get predictions
intensities = np.asarray(dl.run(structure_model, X))
phases = np.asarray([dl.run(phase_model, X)]*2*np.pi-np.pi)
```

DLHub supplies both computational environment and resources.
we then register the container and the function `dlhub_run()` with `funcX`.

```python
from home_run import create_servable
with open("dlhub.json") as fp:
    shim = create_servable(json.load(fp))
```

dlhub.json contains all servable-specific info.
DLHub Use Case Examples

**X-Ray Science**
- Predict structure and phase of a material given coherent diffraction intensity
- Data available from Github

**Energy Storage**
- Predict molecular energies with G4MP2 accuracy at B3LYP cost
- Data available in MDF

**Tomography**
- Enhance tomographic scans and remove noise using generative adversarial model
- Example data available on Petrel

---

**Tomogan: Low-Dose X-Ray Tomography with Generative Adversarial Networks**
Zengchun Liu, Tekin Bicer, Rajkumar Kettimuthu, Doga Gursoy, Francesco De Carlo, Ian Foster
Foundry Concept

• Radically reduce the energy barrier to access curated ML datasets and ML models
• Facilitate reuse, meta-studies, benchmarking, and more
• Long term implications for education

From foundry import Foundry
f = Foundry()
X, y = f.load("dataset1", v="1.0")
y_pred = f.run("model1", v="1.0", X)

• Models run locally or on distributed endpoints
• Capabilities to pull datasets to desired location or move compute to desired location

• Radically reduce the energy barrier to access curated ML datasets and ML models
• Facilitate reuse, meta-studies, benchmarking, and more
• Long term implications for education

From foundry import Foundry
f = Foundry()
X, y = f.load("dataset1", v="1.0")
y_pred = f.run("model1", v="1.0", X)

• Models run locally or on distributed endpoints
• Capabilities to pull datasets to desired location or move compute to desired location

(Dane Morgan, Paul Voyles, Michael Ferris, Marcus Schwarting, Aristana Scourtas, KJ Schmidt, Ben Blaiszik)

NSF CSSI Started Oct. 2019
Thank You!

This work was supported in part by Laboratory Directed Research and Development funding from Argonne National Laboratory under U.S. Department of Energy under Contract DE-AC02-06CH11357. We also thank the Argonne Leadership Computing Facility for access to the PetrelKube Kubernetes cluster and Amazon Web Services for providing research credits to enable rapid service prototyping. This research used resources of the Argonne Leadership Computing Facility, a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

This work was supported by the National Science Foundation under NSF Award Number: 1931306 "Collaborative Research: Framework: Machine Learning Materials Innovation Infrastructure".

Integrative Materials and Design

This work was supported by the National Science Foundation as part of the Midwest Big Data Hub under NSF Award Number: 1636950 “BD Spokes: SPOKE: MIDWEST: Collaborative: Integrative Materials Design (IMaD): Leverage, Innovate, and Disseminate.”

Contact: Aristana Scourtas (aristana@uchicago.edu)
Ben Blaiszik (blaiszik@uchicago.edu)
Backup Slides
What are FAIR Data Principles?

- Findable
- Accessible
- Interoperable
- Reusable

Set of principles to help make data as useful as possible to the community

https://www.force11.org/group/fairgroup/fairprinciples

What is the state of FAIR data and ML in materials science?
FAIR Data Principles

Findable

- Data have an identifier
- Data are registered in a searchable resource

Accessible

- Data accessible via identifier
- Data retrievable by open protocols
FAIR Data Principles

Interoperable

- Data leverage formalized shared vocabularies
- Vocabularies themselves follow FAIR principles

Reusable

- Clear licensing
- Descriptive metadata is sufficient to promote reuse
The Materials Data Facility (MDF)

- Connect: Extract domain-relevant metadata / transform the data
- Publish: Built to handle big data (many TB, millions of files), provides persistent identifier for data, distributed storage enabled
- Discover: Programmatic search index to aggregate and retrieve data across hundreds of indexed data sources

> 35 TB of data
> 320 published authors
> 400 datasets

https://www.materialsdatafacility.org
The Materials Data Facility
DATA AND LEARNING HUB FOR SCIENCE (DLHUB)

• Collect, publish, categorize models and pre/post processing code

• Operate models as a service to simplify sharing, consumption, and access

• Identify models with unique and persistent identifiers (e.g., DOI)

• Implement versioning, search, access controls etc.

Goal: Deliver FAIR for ML
DLHub – A Data and Learning Hub for Science

Describe
- Specify the model files
- Mark up the model with information to make it discoverable and usable

```python
from dlhub_sdk.models.servables.keras import KerasModel
m = KerasModel.create_model("plbl-example.h5")
m.set_title("CANDLE Pilot 1 - Benchmark 1")
m.set_name("candle_plbl")
m.set_domains("genomics","biology"","HPC")
```

Publish
- Register with DLHub for containerization as a servable
- DLHub service creates unique endpoint for servable

```python
from dlhub_sdk.client import DLHubClient
dl = DLHubClient()
dl.publish(m)
```

Discover
- Discover servables with advanced search capabilities through Python SDK or web UI
- Make predictions by sending data to DLHub and specifying the servable to use

```python
from dlhub_sdk.client import DLHubClient
dl = DLHubClient()
pred = dl.run("candle_plbl", data)
```

Run
- Make predictions by sending data to DLHub and specifying the servable to use

DOI for model
Unique endpoint for each model
Search index for discovery
Python tooling
Ability to run models on distributed compute resources
Building on Globus PaaS

- Authentication
- User groups
- Data staging and movement
- Automation capabilities
- Search