Bridges-2 Webinar

Welcoming the NVIDIA AI Enterprise Suite to Bridges-2

Jon Coons, NVIDIA
Julian Uran, PSC

June 18, 2025

Pittsburgh Supercomputing Center

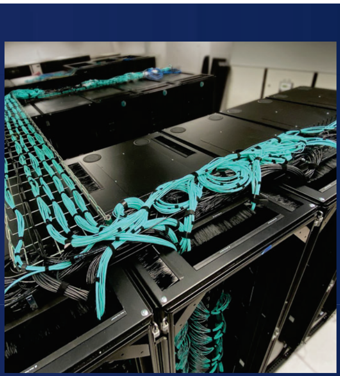
enabling discovery since 1986

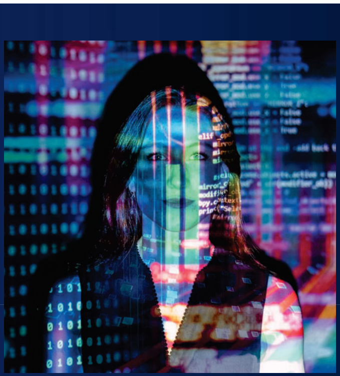
The **Pittsburgh Supercomputing Center (PSC)** provides advanced research computing capability, education, and expertise to the national research community.

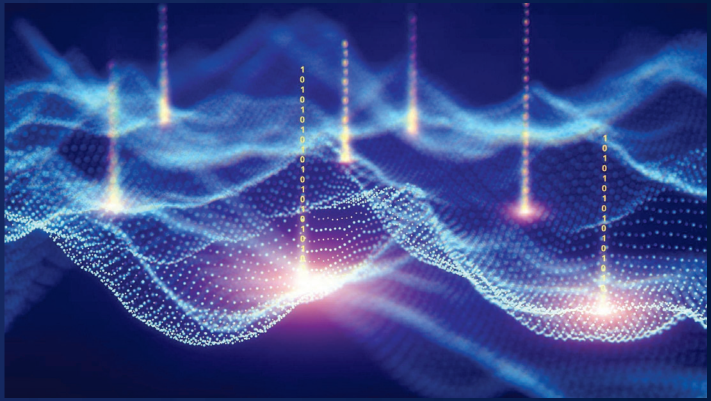
Since 1986, PSC has provided university, government, and industry researchers with access to some of the most powerful systems available for high-performance computing, enabling discovery across all fields of science.

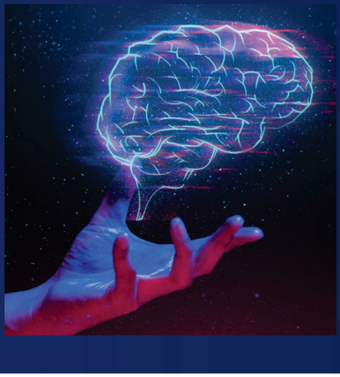
OUR AREAS OF EXPERTISE

- high-performance and data-intensive computing
- data management technologies
- software architecture, implementation, and optimization
- enabling ground-breaking science, computer science, and engineering
- user support for all phases of research and education
- STEM outreach in data science, bioinformatics, and coding









Welcome!

BRIDGES-2

PITTSBURGH SUPERCOMPUTING CENTER
Hewlett Packard Enterprise
NSF
Carnegie Mellon University
Bridges-2
Bridges-2 Overview

Hewlett Packard Enterprise is delivering *Bridges-2*

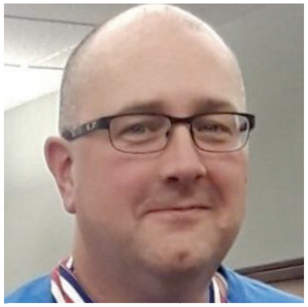
Bridges-2 Leadership Team

Sergiu Sanielevici
PI & Dir. Support
for Sci. Apps.

Robin Scibek
Dir. Comms.
co-PI

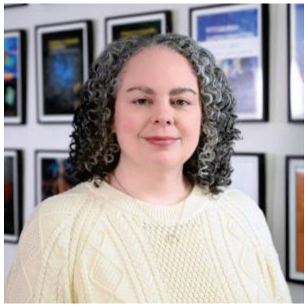
Paola Buitrago
Dir. AI & Big Data
co-PI

Edward Hanna
Dir. Systems & Ops.
co-PI



Tom Maiden
User Services Mgr.
co-PI

Riaz Khatri
Project Manager



Joanne Peca
Information Security Officer

Bridges-2 Webinars

- A forum for the Bridges-2 community to learn and share ideas and achievements: Bridges-2 Webinar series | PSC
- Topics and speakers of interest to work that is being done, or that may be done in future.
- Please suggest future speakers (including from your own team) and/or topics (including your own)!

Just email: sergiu@psc.edu

Introducing today's presenters

Jon Coons is a key member of the NVIDIA AI Enterprise software team, serving as a Senior AI Product Specialist for Generative AI and AI Inferencing for the Americas. Prior to NVIDIA, Jon was an integral part of the AI Global Black Belt team for Microsoft Azure, working with primarily Fortune 500 Enterprises on Generative AI, Vision/Video Analytics and Edge AI initiatives. Bridging the divide of technical acumen and foundational business logic, Jon leverages his diverse background to bring a unique and valuable perspective to the Generative AI equation.

Julian Uran is a Senior Systems Software Engineer at the Pittsburgh Supercomputing Center, where he works on the AI and Big Data team. With a master's degree in systems and computer engineering, Julian is a go-to expert for all things tech—software development, advanced troubleshooting, DevOps, virtualization, system tuning, and hardening. He plays a key role in supporting cutting-edge research by helping users maximize the performance of HPC systems like Bridges-2. At this event, Julian will demonstrate how to run workflows with NVIDIA NIM containers on Bridges-2's H100 GPUs, making AI at scale more accessible to the research community.

Q&A Logistics

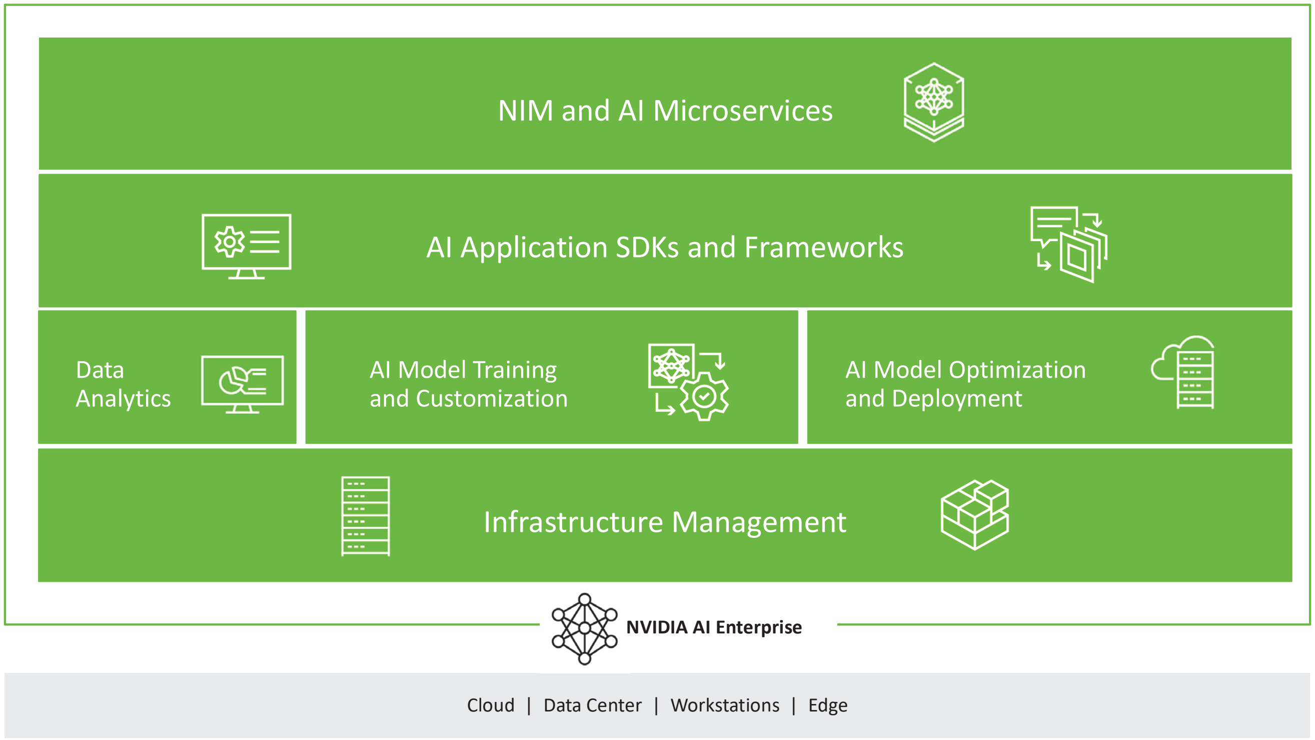
- **We abide by https://support.access-ci.org/code-of-conduct**
- All of us except our speakers will be muted during their presentation.
- Please type your questions into the Zoom chat.
- After the presentation, our speakers will answer questions live during the final ~10 minutes of this webinar.
- The video recording of this webinar and the slides will be linked from https://www.psc.edu/events/bridges-2-webinar-series/nvidia-ai-enterprise-suite/ next week.

NVIDIA AI Enterprise Overview – PSC

Jon Coons, Senior Product Specialist

NVAIE | Generative AI & Inference

NVIDIA AI Enterprise



NIM and AI Microservices

AI Application SDKs and Frameworks

Data Analytics

AI Model Training and Customization

AI Model Optimization and Deployment

Infrastructure Management

NVIDIA AI Enterprise

Cloud | Data Center | Workstations | Edge

NVIDIA AI Enterprise Supported Frameworks (1/3)

For additional information, see NGC Catalog

Framework	Description
Maxine	NVIDIA Maxine is a suite of high-performance, easy-to-use, NVIDIA Inference Microservices (NIMs) and SDKs for deploying AI features that enhance audio, video, and augmented reality effects for video conferencing and telepresence.
TAO Toolkit	The open-source NVIDIA TAO, built on TensorFlow and PyTorch, uses the power of transfer learning while simultaneously simplifying the model training process and optimizing the model for inference throughput on practically any platform.
DeepStream	NVIDIA's DeepStream SDK is a complete streaming analytics toolkit based on GStreamer for AI-based multi-sensor processing, video, audio, and image understanding.
Metropolis	NVIDIA Metropolis is an application framework, set of developer tools, and partner ecosystem that brings visual data and AI together to improve operational efficiency and safety across a range of industries.
NeMo	NVIDIA NeMo™ is an end-to-end platform for developing custom generative AI—including large language models (LLMs), multimodal, vision, and speech AI —anywhere.
TensorRT	NVIDIA® TensorRT™ is an ecosystem of APIs for high-performance deep learning inference.
Triton Inference Server	An open-source software that helps standardize model deployment and delivers fast and scalable AI in production.
TensorFlow	TensorFlow is an open-source platform for machine learning. It provides comprehensive tools and libraries in a flexible architecture allowing easy deployment across a variety of platforms and devices

NVIDIA AI Enterprise Supported Frameworks (2/3)

For additional information, see NGC Catalog

Framework	Description
Riva	NVIDIA® Riva is a set of GPU-accelerated multilingual speech and translation microservices for building fully customizable, real-time conversational AI pipelines.
Monai	MONAI is a freely available, community-supported, PyTorch-based framework for deep learning in healthcare imaging.
Clara	NVIDIA Clara is a suite of computing platforms and software services that powers AI healthcare solutions from imaging to genomics and drug discovery.
Clara Holoscan	NVIDIA Holoscan is the sensor processing platform that streamlines the development and deployment of AI and high-performance computing (HPC) applications for real-time insights.
RAPIDS	RAPIDS is a suite of GPU-accelerated data science and AI libraries with APIs that match popular open-source data tools.
cuOpt	NVIDIA® cuOpt™ optimizes operations by enabling better, faster decisions with accelerated computing.
Merlin	NVIDIA Merlin empowers data scientists, machine learning engineers, and researchers to build high-performing recommenders at scale.
Morpheus	NVIDIA Morpheus is an open application framework that supports real-time data monitoring with AI and threat detection in data centers and cloud.

NVIDIA AI Enterprise Supported Frameworks (3/3)

For additional information, see NGC Catalog

Framework	Description
Modulus	NVIDIA Modulus is an open-source framework for building, training, and fine-tuning Physics-ML models with a simple Python interface.
CUDA	The CUDA compute platform extends from the 1000s of general-purpose compute processors featured in our GPU's compute architecture, parallel computing extensions to many popular languages, powerful drop-in accelerated libraries to turnkey applications and cloud-based compute appliances.
CUDA Toolkit	The NVIDIA® CUDA® Toolkit provides a development environment for creating high-performance, GPU-accelerated applications.
HPC SDK	The NVIDIA HPC Software Development Kit (SDK) includes the proven compilers, libraries and software tools essential to maximizing developer productivity.
Kubernetes Device Plugin	The NVIDIA Kubernetes Device Plugin registers GPUs as compute resources in the Kubernetes cluster.
PyTorch Geometric	PyG (PyTorch Geometric) is a library built upon PyTorch to easily write and train Graph Neural Networks (GNNs) for a wide range of applications related to structured data.
PyTorch	PyTorch is a GPU accelerated tensor computational framework.
DGL	Deep Graph Library (DGL) is a Python package built for the implementation and training of graph neural networks on top of existing DL frameworks

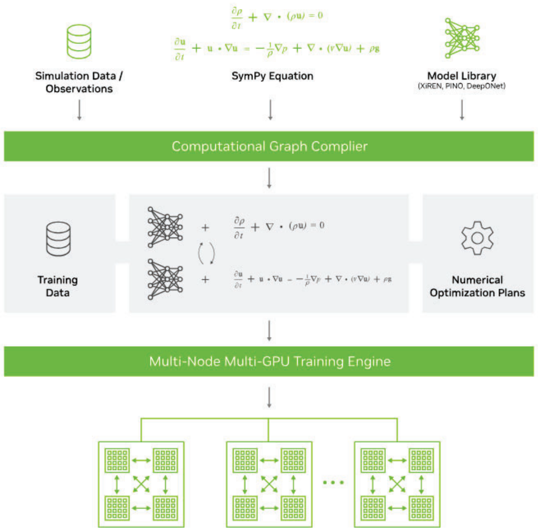
NVIDIA Modulus

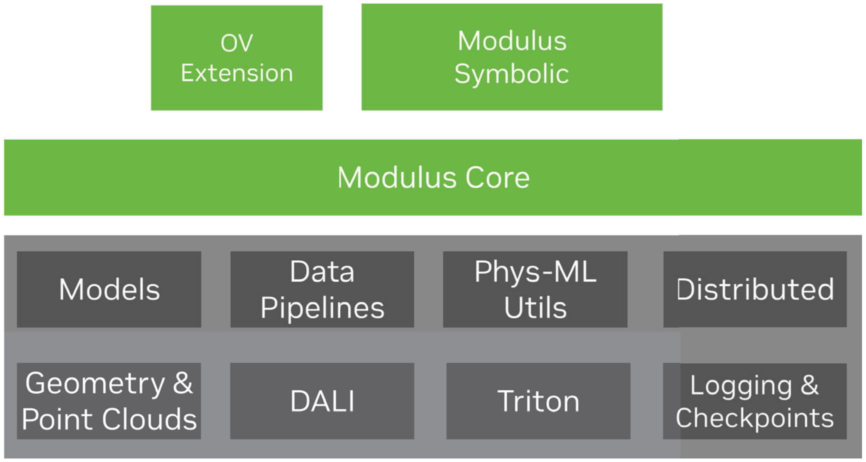
NVIDIA Modulus

Open-Source Platform for developing physics-ml **algorithms**

- Open-Source project for Physics-ML innovation and collaboration
- Platform for building physics-ml models using architectures and algorithms that satisfy first principles

- Training and inference pipelines that bring the best of NVIDIA's AI stack for scalable and optimal performance


$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$
$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \rho \mathbf{g}$
Simulation Data / Observations
SymPy Equation
Model Library
(XiREN, PINO, DeepONet)
Computational Graph Complier
Training Data
+
$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$
+
$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \rho \mathbf{g}$
Numerical Optimization Plans
Multi-Node Multi-GPU Training Engine
...



OV Extension

Modulus Symbolic

Modulus Core

Models

Data Pipelines

Phys-ML Utils

Distributed

Geometry & Point Clouds

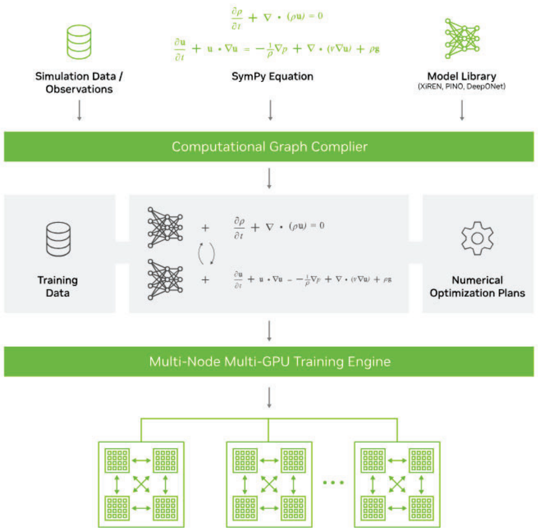
DALI

Triton

Logging & Checkpoints

Open-Source AI Toolkit for Physics AI

- Enterprise grade toolkit for Physics-ML:
  - Python based, designed for coupling data and physics, optimal performance and scalable
- Reference case studies and recipes as starting points
  - Model architecture Zoo
- Research and develop your Physics-ML library:
  - Interoperable with PyTorch – Import your R&D model*
- Facilitates open collaboration within the Physics-ML scientific community

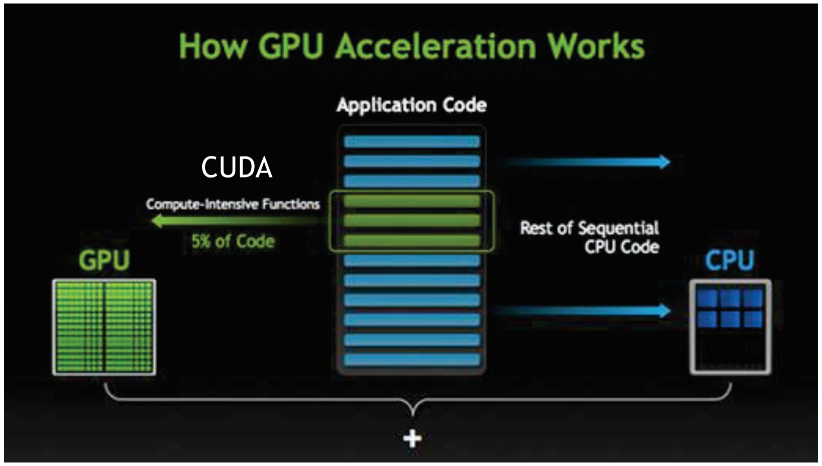


$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$
$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \rho \mathbf{g}$
Simulation Data / Observations
SymPy Equation
Model Library
(XiREN, PINO, DeepONet)
Computational Graph Complier
Training Data
+
$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$
+
$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \rho \mathbf{g}$
Numerical Optimization Plans
Multi-Node Multi-GPU Training Engine
...

Source code: https://github.com/NVIDIA/modulus

What is it not?

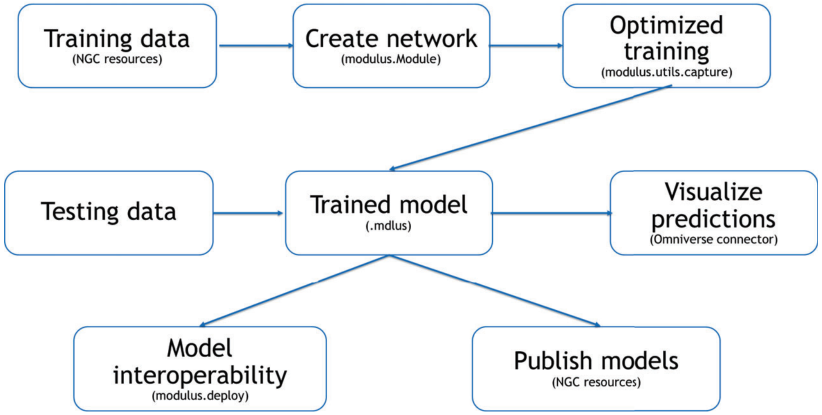
Not a purpose-built app


How GPU Acceleration Works
Application Code
CUDA
Compute-Intensive Functions
5% of Code
GPU
Rest of Sequential CPU Code
CPU
+

No out of the box solutions – Build your solution using Modulus Toolkit

But we do offer reference workflows.

But how to frame my problem as an AI problem?



Training data
(NGC resources)

Create network
(modulus.Module)

Optimized training
(modulus.utils.capture)

Testing data

Trained model
(.mdlus)

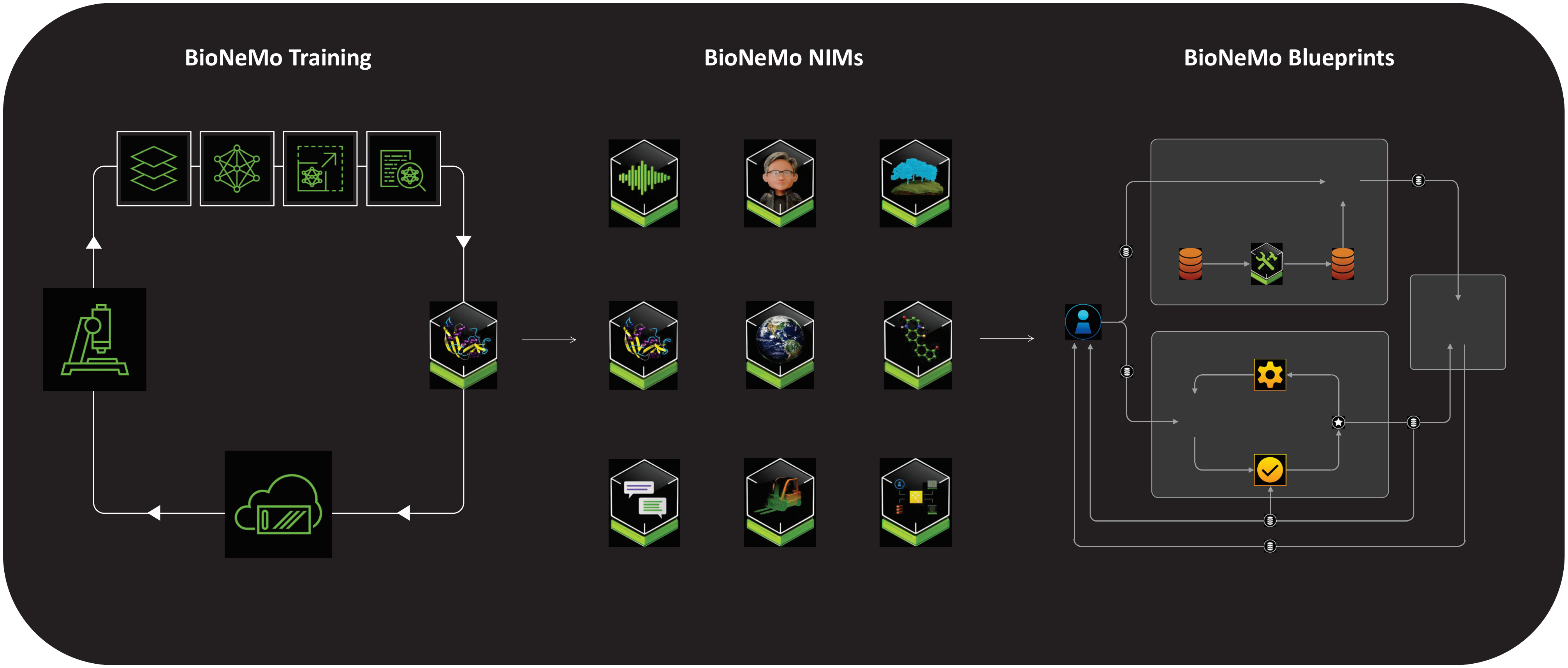
Visualize predictions
(Omniverse connector)

Model interoperability
(modulus.deploy)

Publish models
(NGC resources)

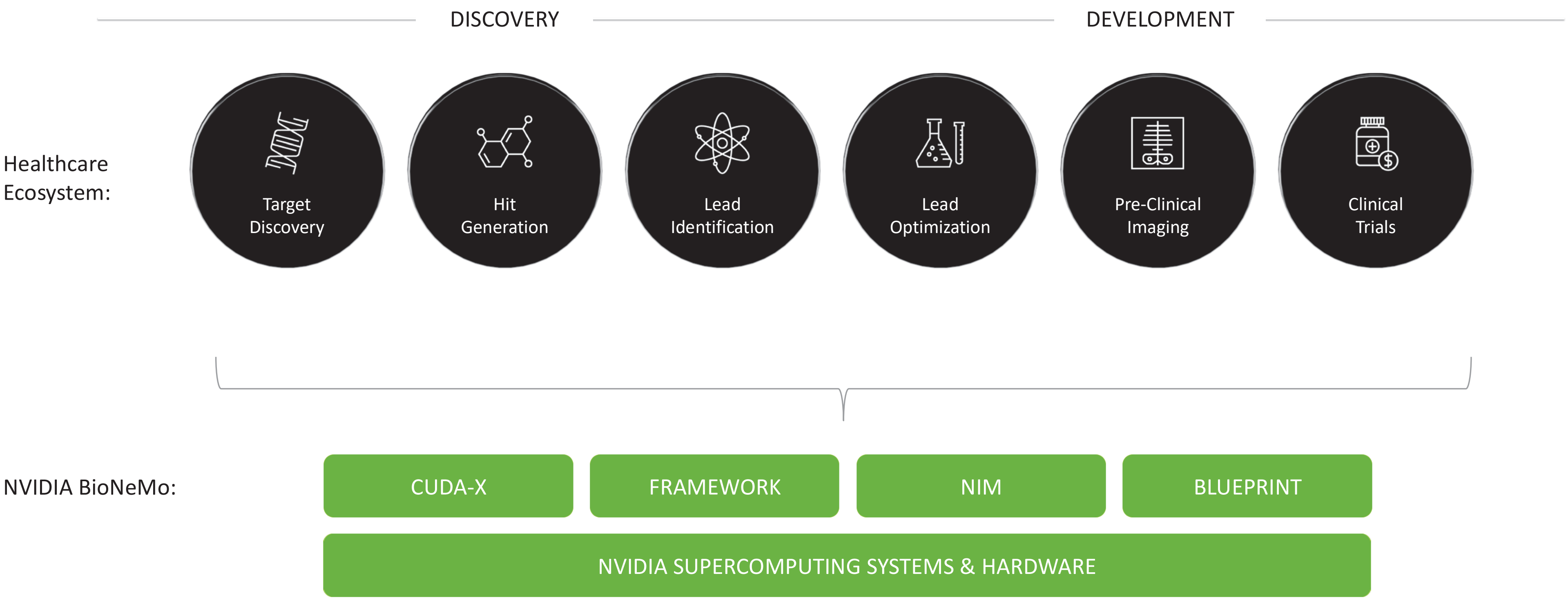
NVIDIA BioNeMo Platform & NIMs

NVIDIA BioNeMo Platform



NVIDIA BioNeMo Platform

Tools to help the healthcare ecosystem build AI-driven drug discovery workflows



DISCOVERY

DEVELOPMENT

Healthcare Ecosystem:

- Target Discovery
- Hit Generation
- Lead Identification
- Lead Optimization
- Pre-Clinical Imaging
- Clinical Trials

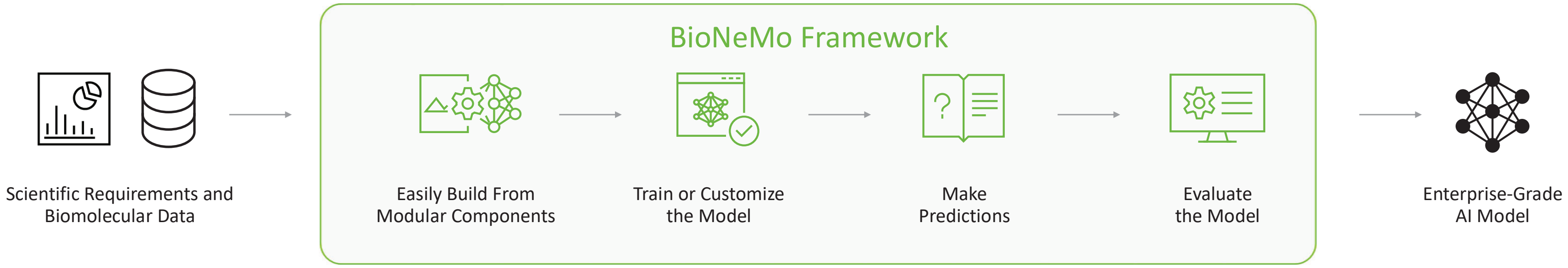
NVIDIA BioNeMo:

- CUDA-X
- FRAMEWORK
- NIM
- BLUEPRINT

NVIDIA SUPERCOMPUTING SYSTEMS & HARDWARE

NVIDIA BioNeMo Framework

Build Better Models Faster for Drug Discovery, Now Open-Source on GitHub



Scientific Requirements and Biomolecular Data

BioNeMo Framework

Easily Build From Modular Components

Train or Customize the Model

Make Predictions

Evaluate the Model

Enterprise-Grade AI Model

Full Optimization

Easy to Use

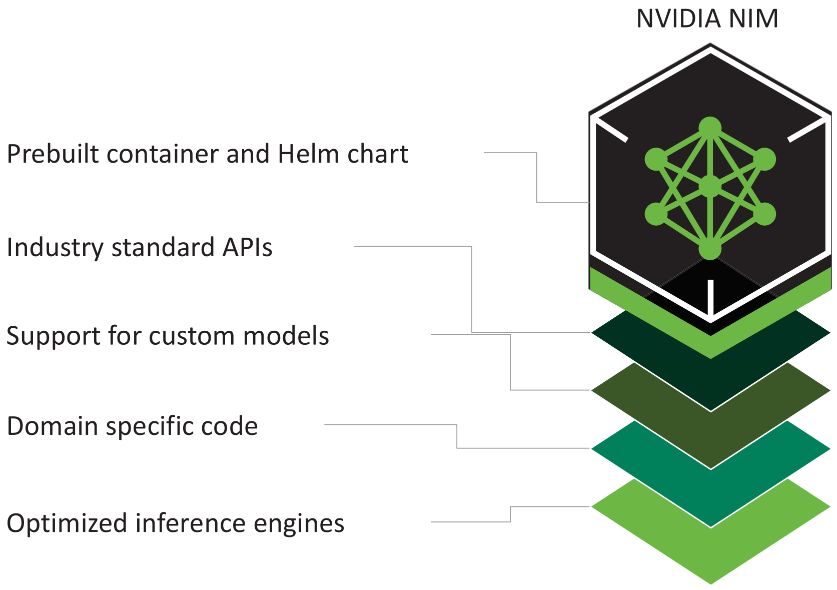
Enterprise-grade

Customizability

Reference: https://arxiv.org/abs/2411.10548

NVIDIA NIM: Inference Microservices for Generative AI

Accelerated runtime for generative AI


NVIDIA NIM
Prebuilt container and Helm chart
Industry standard APIs
Support for custom models
Domain specific code
Optimized inference engines

Deploy anywhere and maintain control of generative AI applications and data

Simplified development of AI application that can run in enterprise environments

Day 0 support for all generative AI models providing choice across the ecosystem

Improved TCO with best latency and throughput running on accelerated infrastructure

Best accuracy for enterprise by enabling tuning with proprietary data sources

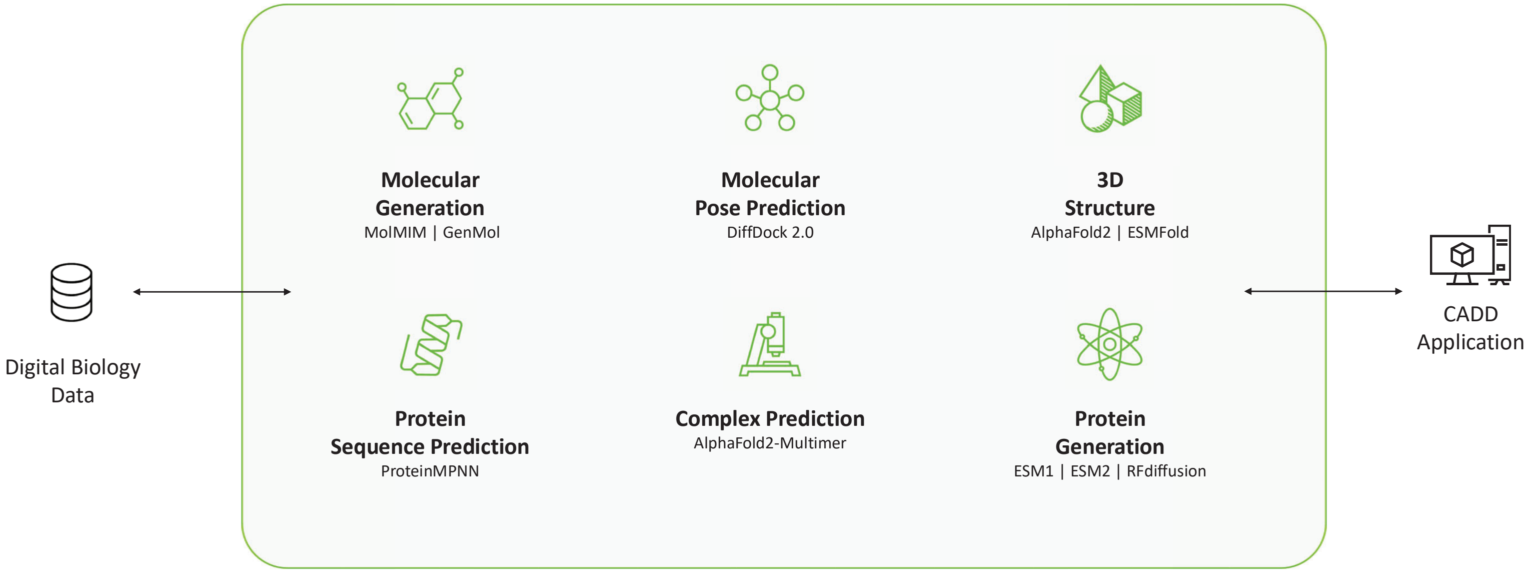
Enterprise software with feature branches, validation and support

Microsoft Azure   aws   Google Cloud   ORACLE   DGX & DGX Cloud   DELL Technologies   Hewlett Packard Enterprise   Lenovo   SUPERMICRO

NVIDIA

NVIDIA BioNeMo NIM microservices

Instantly use your own data with optimized biomolecular models deployable anywhere



Digital Biology Data

Molecular Generation
MolMIM | GenMol

Molecular Pose Prediction
DiffDock 2.0

3D Structure
AlphaFold2 | ESMFold

Protein Sequence Prediction
ProteinMPNN

Complex Prediction
AlphaFold2-Multimer

Protein Generation
ESM1 | ESM2 | RFdiffusion

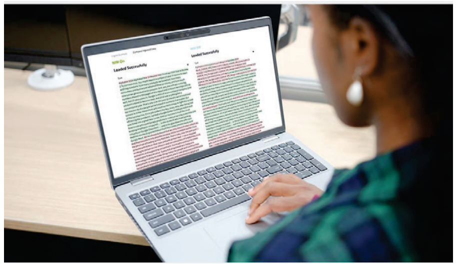
CADD Application

NVIDIA BioNeMo Blueprints

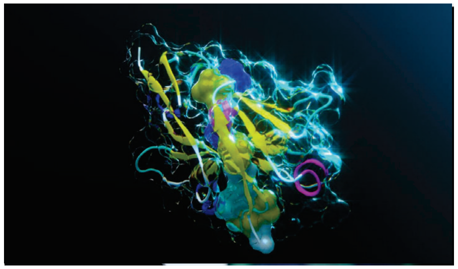
NVIDIA BioNeMo Blueprints Are Reference Workflows for Drug Discovery

Available on build.nvidia.com

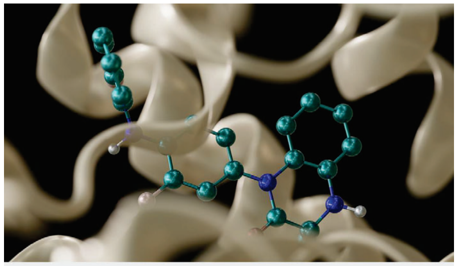
Multimodal PDF Data Extraction for Enterprise RAG



Generative Protein Binder Design for Drug Discovery



Generative Virtual Screening for Drug Discovery



• • •

monthly release

NVIDIA NIM Agent Blueprint

Example Application

Interactive experience that can be easily replicated

Sample Data

Public data for workflow testing

Reference Code

Reference code for constructing workflows

Architecture

Reference architecture including API definitions, NIM, and more

Customization Tools

Customize and evaluate models

Orchestration Tools

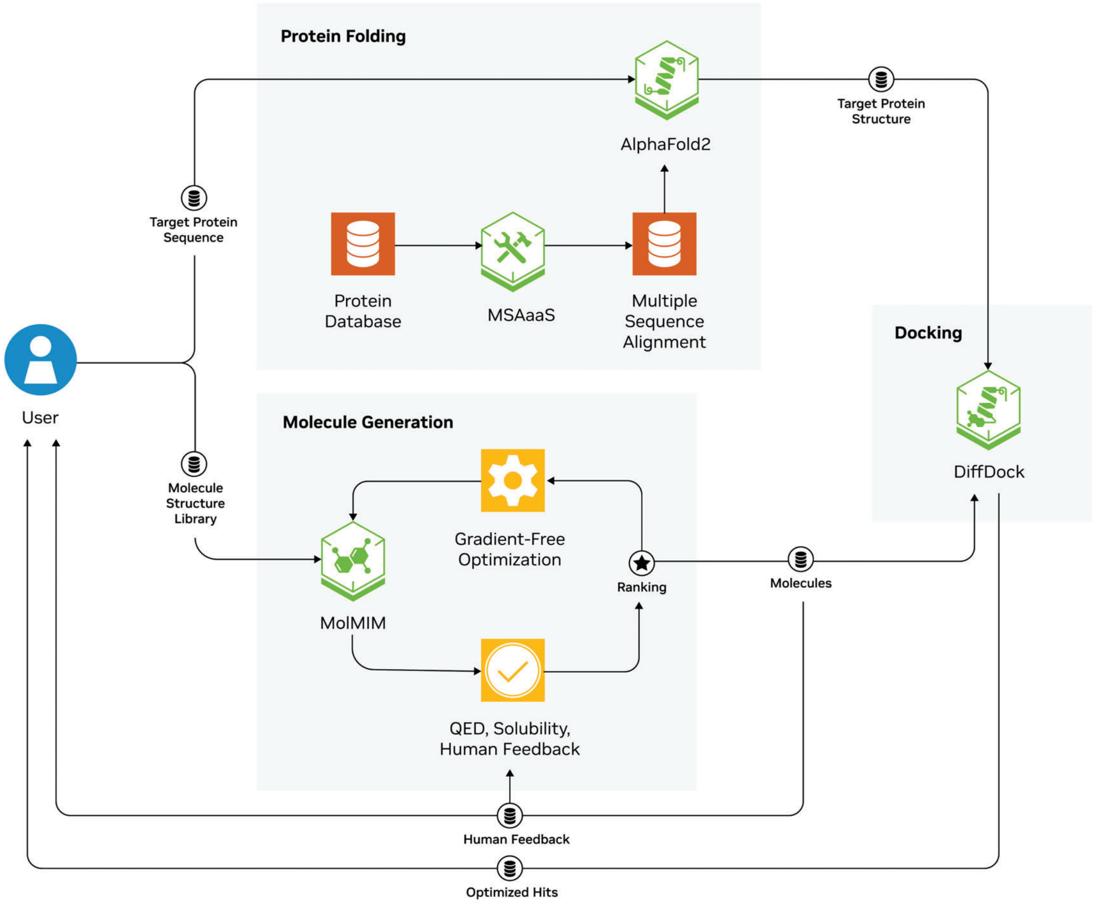
Deploy and manage workflow microservices

Generative Virtual Screening

Models: MSA, AlphaFold2, MolMIM, DiffDock 2.0

Benefits

- Use generative AI to more efficiently explore chemical space to optimize molecular designs for multiple features simultaneously
- Accelerated NIMs allow rapid evaluation of large molecule databases to identify better drug candidates faster
- Test fewer molecules to identify virtual hits, reducing the time and cost of drug development



Generative Protein Binder Design

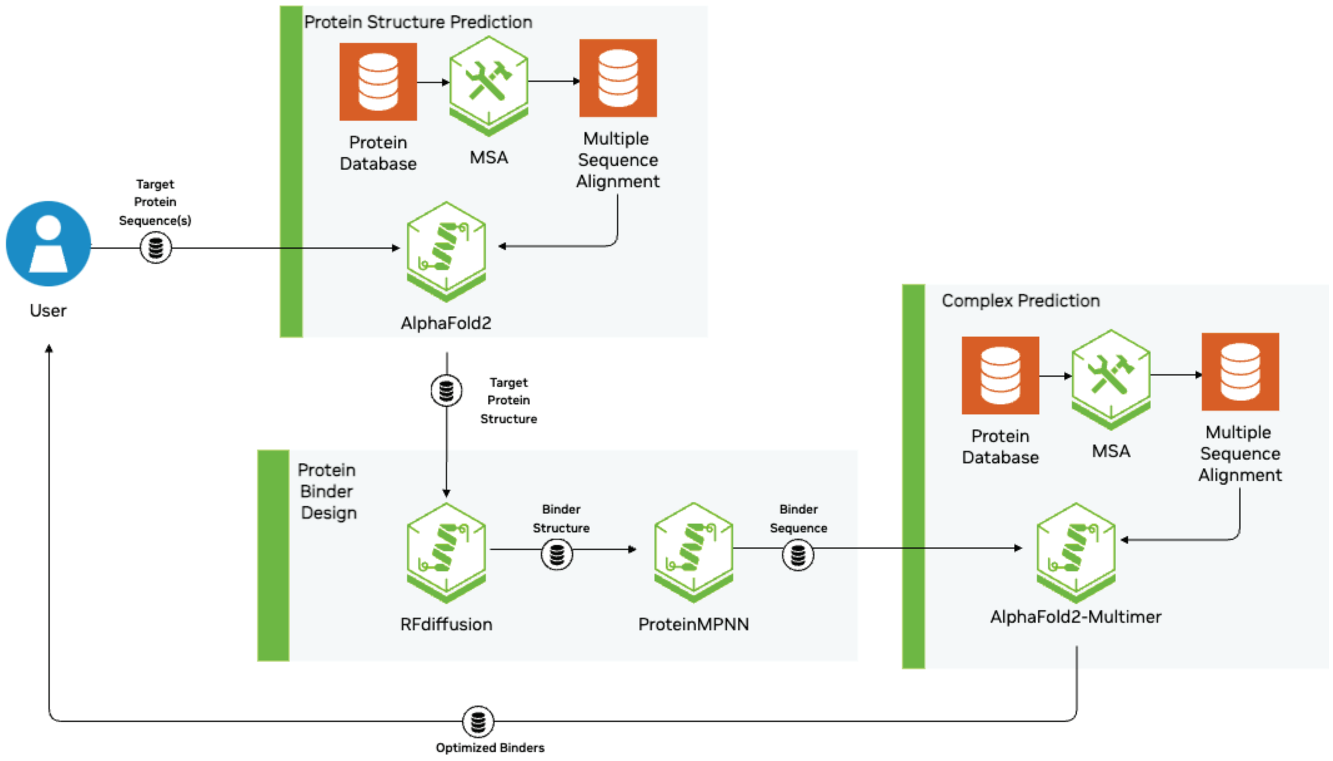
Models: MSA, AlphaFold2, RFdiffusion, ProteinMPNN, Alphafold2-Multimer

Use Cases

- Design novel proteins to bind to a target of interest and/or perform a desired function.

Value Prop:

- Efficiently explore vast protein design space for structures with precise functional features
- SOTA speed and scalability over other approaches
- Identify high-affinity binders from a smaller set of designs



Thank you!

Appendix

NVIDIA Clara for Drug Discovery and Development

A Computer-Aided Drug Discovery Acceleration Platform

DISCOVERY

DEVELOPMENT

TARGET DISCOVERY

HIT GENERATION

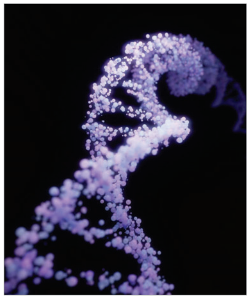
LEAD IDENTIFICATION

LEAD OPTIMIZATION

ANIMAL MODELS

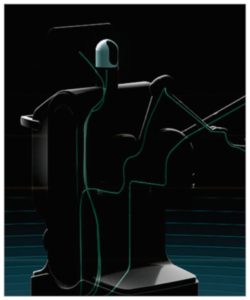
CLINICAL TRIALS

NVIDIA CLARA

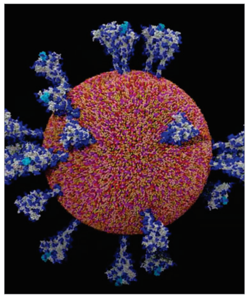


PARABRICKS
Genomics

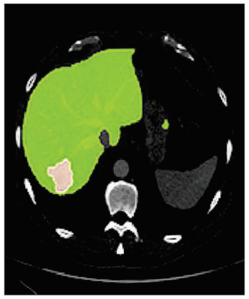
ISAAC
Robotics



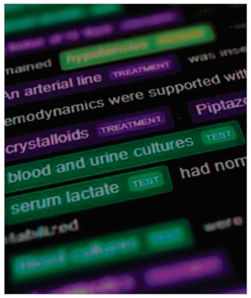
HOLOSCAN
Instruments



BIONEMO
Therapeutics



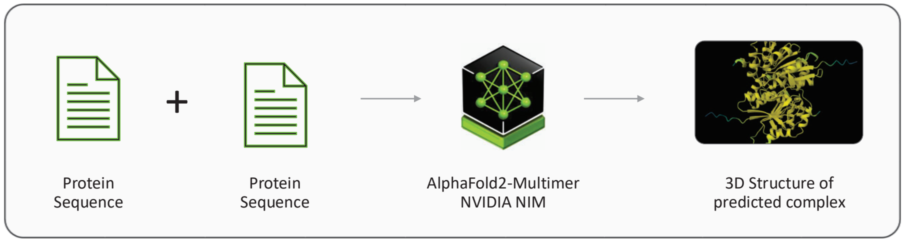
MONAI
Imaging


nained
An arterial line TREATMENT was inse
emodynamics were supported wit
crystalloids TREATMENT Piptaz
blood and urine cultures TEST
serum lactate TEST had nor

NEMO
Natural Language

AlphaFold2-Multimer NIM

a major advancement in protein complex structure prediction


Protein Sequence
+
Protein Sequence
AlphaFold2-Multimer NVIDIA NIM
3D Structure of predicted complex

- **What does it do?**
  - Given two amino acid sequences, predict the 3D structure of those two proteins binding each other.

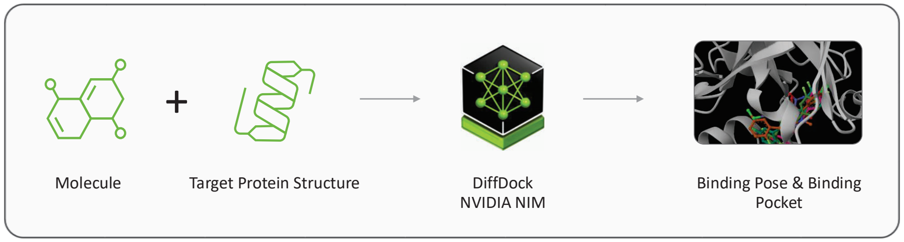
- **Why this NIM?**
  - Reference Accuracy: Outperforms previous SOTA protein complex prediction methods
  - High Demand: Understanding protein structure is a important part of understanding its function and role in health and disease

- **Which blueprints are enabled?**
- Protein Binder Design
  - *The process of creating or discovering molecules that selectively bind a target protein with high affinity and specificity*

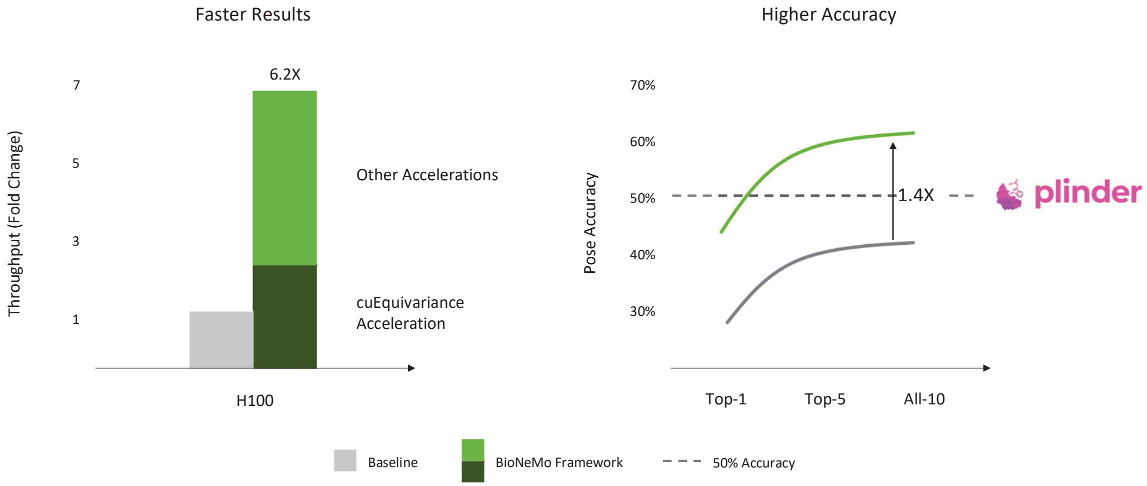
Alphafold2-Multimer: https://www.nature.com/articles/s41586-021-03819-2

DiffDock 2.0 NIM

Molecule protein docking inspired by DiffDock, accelerated and trained by NVIDIA



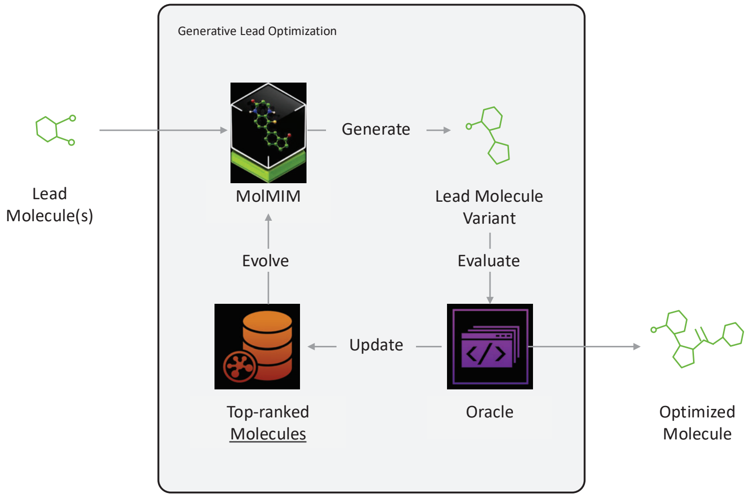
Molecule + Target Protein Structure → DiffDock NVIDIA NIM → Binding Pose & Binding Pocket



DiffDock: https://arxiv.org/abs/2210.01776

MolMIM NIM

Make your chemical design search more targeted and efficient



What does it do?

- Controlled molecular generation
- Multiparameter Optimization
- Accepts User-Defined Oracles

Why this NIM?

- **Highly Customizable:** Tailor molecule generation to specific research needs

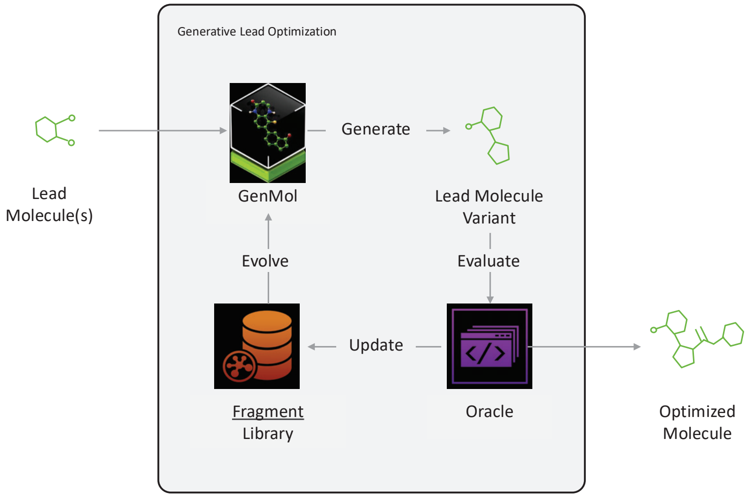
Workflows Enabled

- Virtual screening
  - *Rapid computational evaluation of large chemical libraries to identify potential hits that can bind to a target*
- Lead optimization
  - *Improve molecular hits from initial experiments to improve biochemical qualities needed for a drug (e.g. low toxicity, proper distribution)*

MolMIM: https://arxiv.org/abs/2208.09016

GenMol NIM: a generalist molecular generation model

a generalist foundation model for molecular generation



- **What does it do?**
  - *De novo* or goal-directed molecular generation
  - Accepts User-Defined Oracles and/or fragment libraries

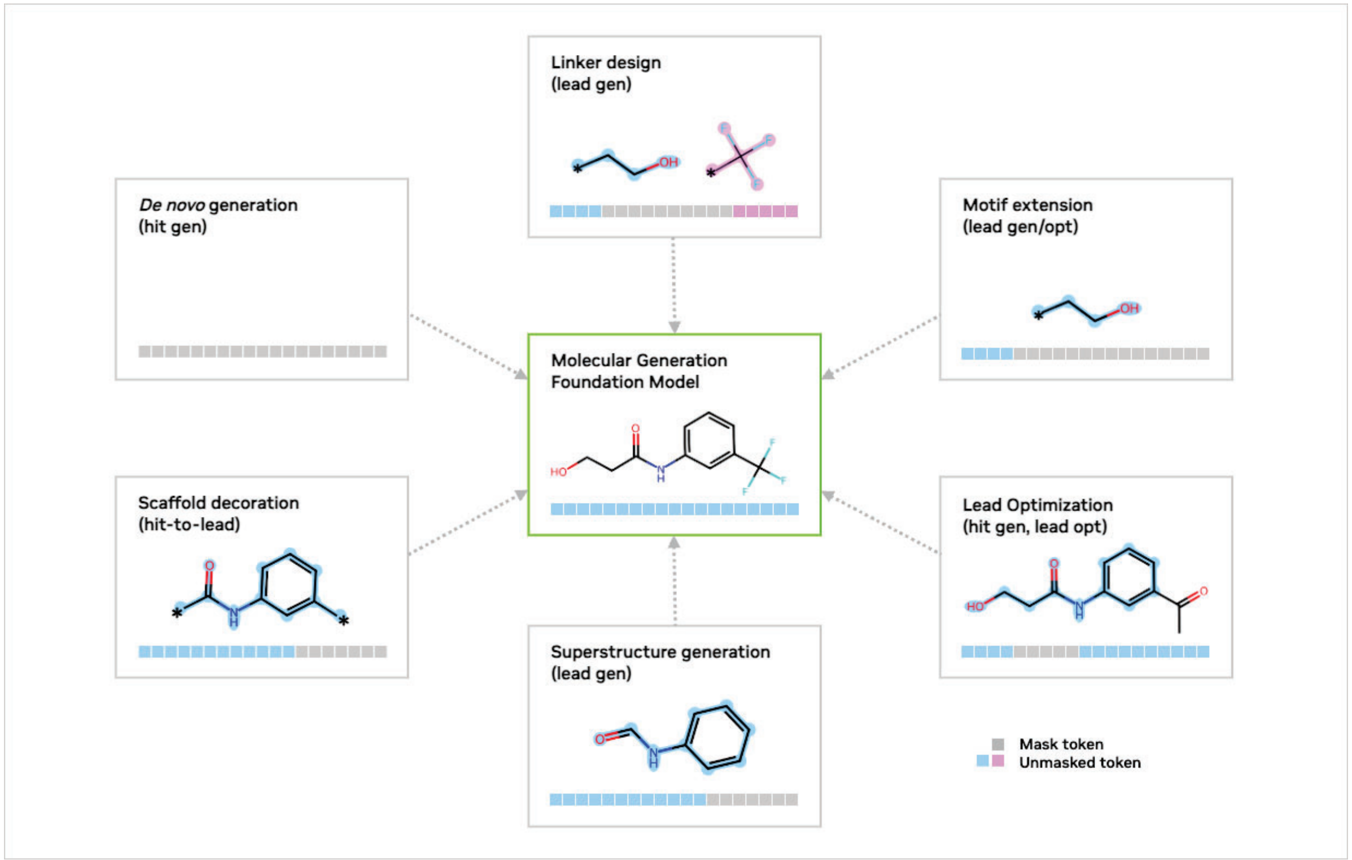
- **Why this NIM?**
  - **Versatility**: SOTA for goal-directed optimization and high performance at diverse molecular tasks without task-specific fine-tuning.
  - **Efficiency**: Up to 35% faster generation compared to traditional models.

- **Workflows Enabled**
- Virtual screening
  - *Rapid computational evaluation of large chemical libraries to identify potential hits that can bind to a target*
- Lead optimization
  - *Improve molecular hits from initial experiments to improve biochemical qualities needed for a drug (e.g. low toxicity, proper distribution)*

GenMol: https://arxiv.org/abs/2501.06158

GenMol Uses SAFE Molecules For a Variety of Tasks

Goal-Directed Molecular Generation for Hit to Lead Workflows



Linker design (lead gen)

De novo generation (hit gen)

Motif extension (lead gen/opt)

Molecular Generation Foundation Model

Scaffold decoration (hit-to-lead)

Lead Optimization (hit gen, lead opt)

Superstructure generation (lead gen)

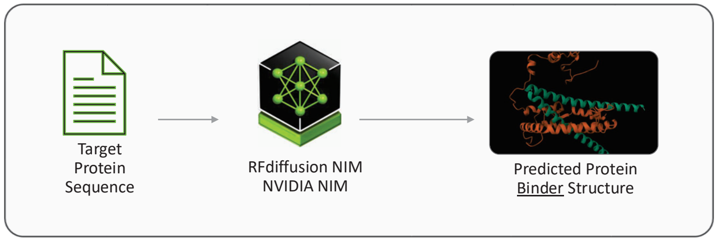
■ Mask token

■■ Unmasked token

GenMol: https://arxiv.org/abs/2501.06158

RFdiffusion NIM

a leading model in guided protein generation


Target Protein Sequence
RFdiffusion NIM NVIDIA NIM
Predicted Protein Binder Structure

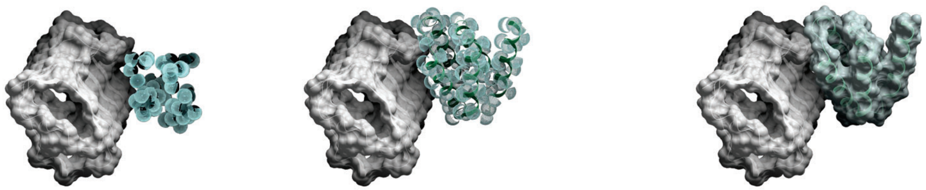
- **What does it do?**
  - Guided, conditional protein structure prediction
  - Unconditional protein structure generation
- **Why this NIM?**
  - **State-of-the-Art Protein Design via Diffusion:** more accurate and flexible *de novo* protein designs compared to many traditional or purely sequence-focused models.
  - **Proven Experimental Validation:** designed proteins have been shown to fold and function as intended—an important advantage for researchers seeking real-world impact.
  - **Seamless Integration and Community Support:** Open-source model with an active user community, offering resources, collaborative tools, and benchmarks
- **Which blueprints are enabled?**
- Protein Binder Design
  - *The process of creating or discovering molecules that selectively bind a target protein with high affinity and specificity*

RFdiffusion: https://www.nature.com/articles/s41586-023-06415-8

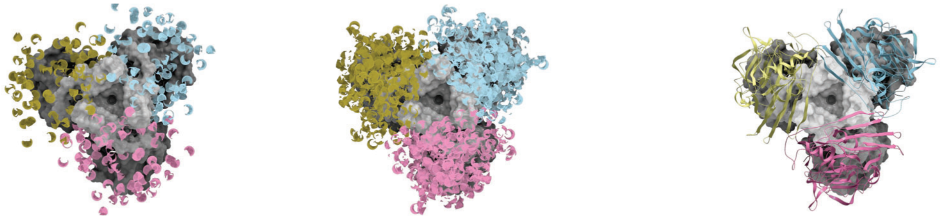
RFdiffusion NIM

a leading model in guided protein generation

Protein Binder Design



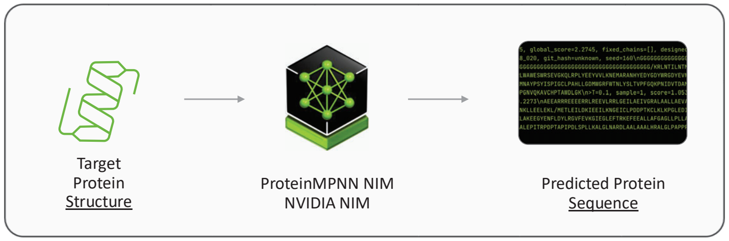
Protein Scaffolding



RFdiffusion: https://www.nature.com/articles/s41586-023-06415-8

ProteinMPNN NIM

a leading model in guided sequence generation



```
5, global_score=2.2745, fixed_chains=[], designe
8_020, git_hash=unknown, seed=160\nGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG/KRLNTILNT
LWAWESWRSEVGKQLRPLYEEYVVLKNEMARANHYEDYGDYWRGDYEV
MNAYPSYISPIGCLPAHLLGDMWGRFWTNLYSLTVPFGQKPNIDVTDA
PGNVQKAVCHPTAWDLGK\n>T=0.1, sample=1, score=1.053
.2273\nAEEARRREEEERRLREEVLRRLGEILAEIVGRALAALLAEVA
NKLLEELEKL/METLEILDKIEEILKNGEICLPDDPTKCLKLKPGLED
LAKEEGYENFLDYLRGVFEVKGIEGLEFTRKEFEEALLAFGAGLLPLLA
ALEPITRPDPTAPIPDLSPLLKALGLNARDLAALAAALHRALGLPAPP
```

- **What does it do?**
  - ○ Given a protein structure, design a sequence that will fold into it
- **Why this NIM?**
  - **High Speed and Scalability:** The model is computationally efficient, enabling the rapid design of thousands of sequences for a single protein backbone. This scalability is ideal for large-scale protein engineering projects.
  - **Experimentally Validated Performance:** Has high experimental success rates in generating sequences that fold into desired structures and maintain functionality
- **Which blueprints are enabled?**
- Protein Binder Design
  - *The process of creating or discovering molecules that selectively bind a target protein with high affinity and specificity*

ProteinMPNN: https://www.science.org/doi/10.1126/science.add2187

Running NVIDIA NIMs on Bridges-2

Two ways to get your AI jobs done, with minimal setup.

Julian Uran
Senior Systems Software Engineer /
Machine Learning Research Engineer, PSC

June 18th, 2025

🧩 Context (Goal & Problem)

What You Want to Do:

- Use NVIDIA’s prebuilt NIMs for AI tasks
- Do it on Bridges-2, with minimal setup

The Challenge:

- 80+ NIMs exist (as of now)
- You want an easy way to run them
- You don’t want to build containers from scratch

✅ Solution Overview

Bridges-2 Supercomputer Resources:

- Nodes with H100 GPUs (V100 not supported for NIMs)
- Reproducible batch/interactive workflows, Slurm scheduling system.

Two Ways to Run NIMs on Bridges-2:

1. 🧾 **Simple Batch Mode:** specify input, run script, get output.
2. 🖥️ **Interactive Mode**: launch server, send custom queries.

✅ Solution Overview

🧠 Choose what fits your workflow.

📦 No need to build, just request NIMs we do not yet have.

📘 Tutorial: Step-by-Step

Learning by doing.

Batch Mode (Simple)

✅ Best for: easy, repeatable jobs (fixed pipelines).

💡 Just add queries to a file and go.

Use Case: I have a list of inputs and just want results.

Steps:

1. `cd /ocean/containers/nim/<container_name>/`
2. Check usage notes: `cat README.md`
3. Add your queries to `input_queries.txt`
4. Run: `sbatch nim_auto.sbatch input_queries.txt output.txt`

Outputs: Will appear in the output folder defined in script.

Interactive Mode

✅ Best for: custom logic, testing.

🖥️ Start server, then query it from client.

Use Case: I want to run my own logic/client against a live API.

Steps:

1. Start interactive Slurm session
2. `cd /ocean/containers/nim/<container_name>/`
3. Run server: `bash run.sh`
4. In another terminal: `bash nim_client.sh` (or use `nim_client.py`)

📚 Files You’ll See

Each NIM folder includes:

File / Folder	Purpose
`nim_auto.sbatch`	Starts server + client batch
`nim_client.sbatch`	Batch client to send queries
`input_queries.txt`	Your input file
`run.sh`	Interactive server launcher
`nim_client.sh/py`	Query interactively
`cache/`	Pre-loaded models, speeds up run
`README.md`	Usage guide tailored to Bridges-2

🎬 Step by Step “Demo”

Let’s See It in Action

I’ll show both workflows using a sample NIM.We’ll cover:

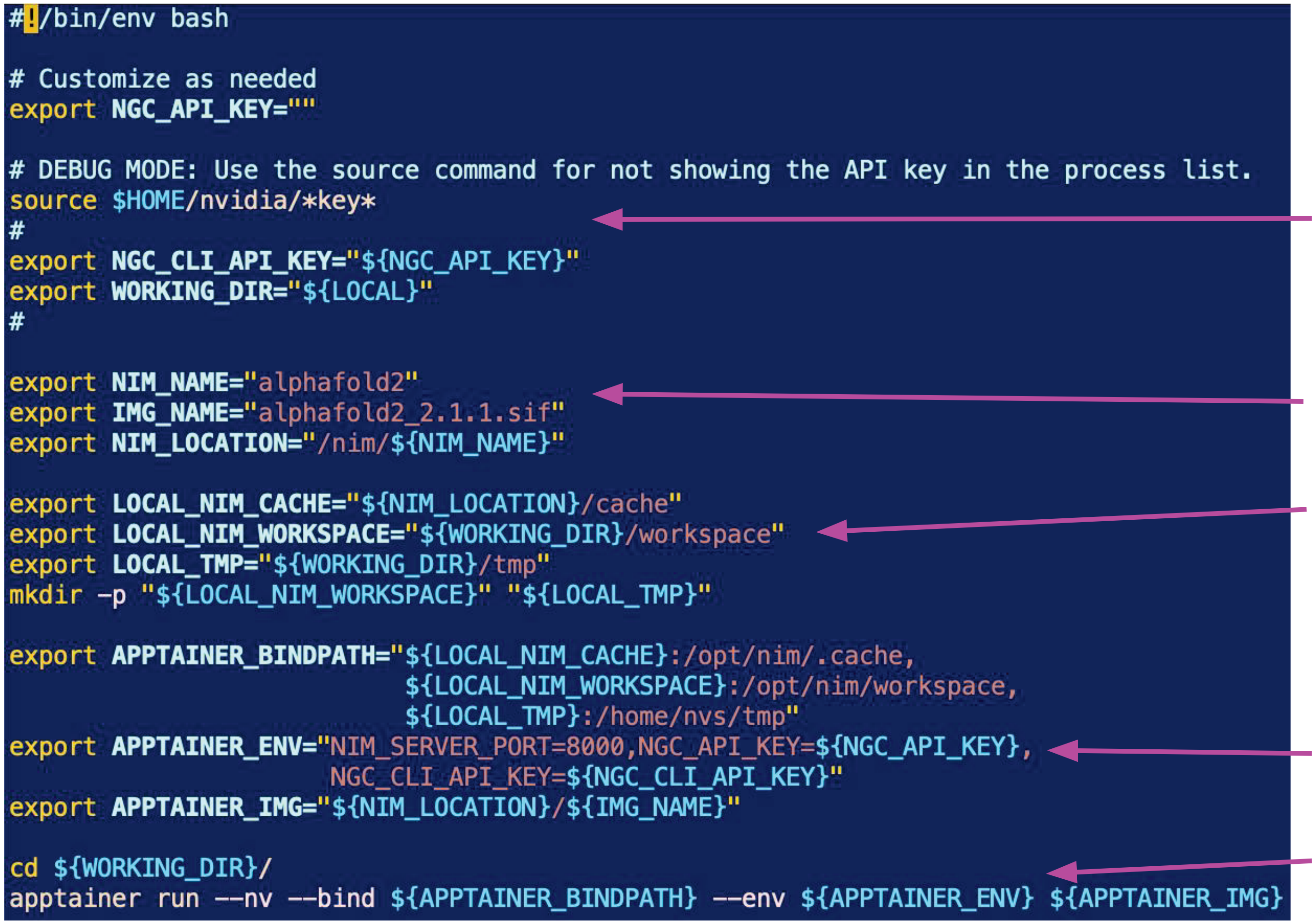
- How to navigate the container folder
- Where to add input
- How to run batch vs interactive

🎬 Interactive: Start Slurm job

```
researcher@br011:~ $
```

```
interact --partition
GPU-small,GPU-shared
--gres=gpu:h100-80:1

cd /ocean/containers/nim/

cd alphafold2

cd interactive
```

Interactive: run.sh



```
#!/bin/env bash

# Customize as needed
export NGC_API_KEY=""

# DEBUG MODE: Use the source command for not showing the API key in the process list.
source $HOME/nvidia/*key*
#
export NGC_CLI_API_KEY="${NGC_API_KEY}"
export WORKING_DIR="${LOCAL}"
#

export NIM_NAME="alphafold2"
export IMG_NAME="alphafold2_2.1.1.sif"
export NIM_LOCATION="/nim/${NIM_NAME}"

export LOCAL_NIM_CACHE="${NIM_LOCATION}/cache"
export LOCAL_NIM_WORKSPACE="${WORKING_DIR}/workspace"
export LOCAL_TMP="${WORKING_DIR}/tmp"
mkdir -p "${LOCAL_NIM_WORKSPACE}" "${LOCAL_TMP}"

export APPTAINER_BINDPATH="${LOCAL_NIM_CACHE}:/opt/nim/.cache,
                          ${LOCAL_NIM_WORKSPACE}:/opt/nim/workspace,
                          ${LOCAL_TMP}:/home/nvs/tmp"
export APPTAINER_ENV="NIM_SERVER_PORT=8000,NGC_API_KEY=${NGC_API_KEY},
                     NGC_CLI_API_KEY=${NGC_CLI_API_KEY}"
export APPTAINER_IMG="${NIM_LOCATION}/${IMG_NAME}"

cd ${WORKING_DIR}/
apptainer run --nv --bind ${APPTAINER_BINDPATH} --env ${APPTAINER_ENV} ${APPTAINER_IMG}
```

Set internal helper variables

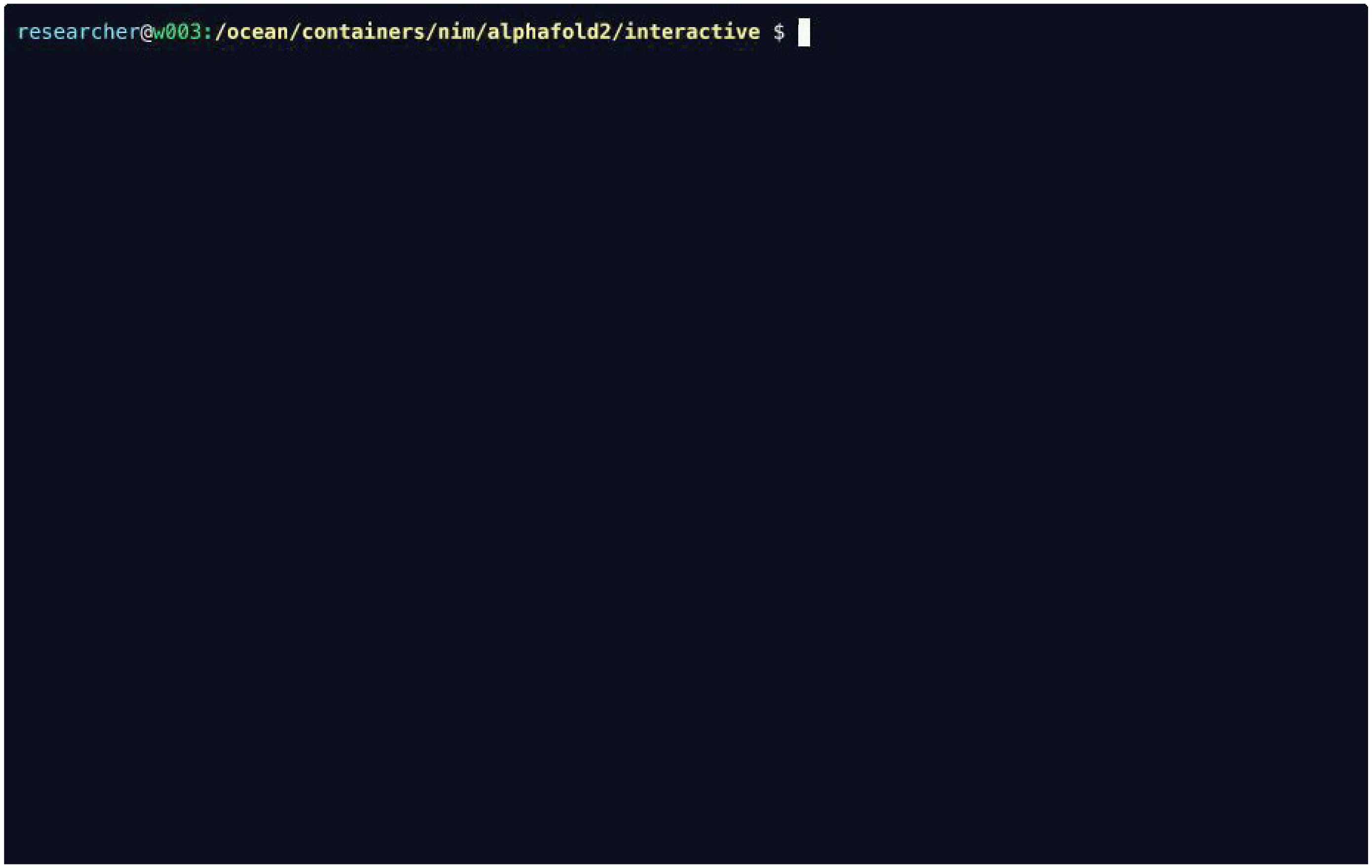
Set container name

Set local disk paths

Set NIM API port

Start up NIM server

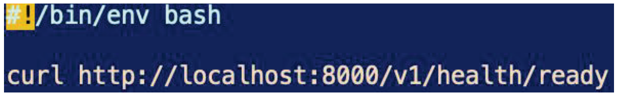
Interactive: run.sh



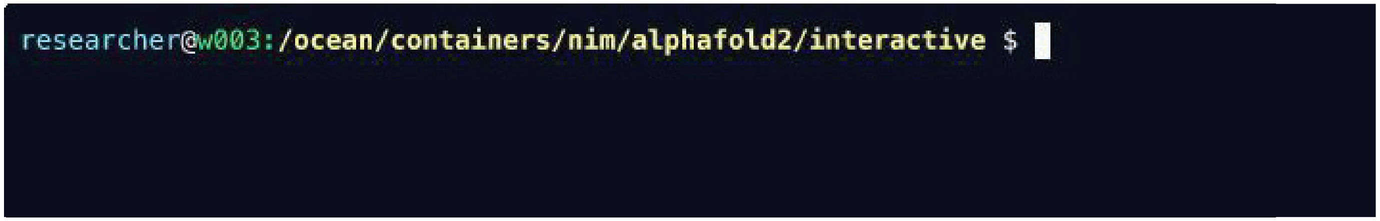
```
researcher@w003:/ocean/containers/nim/alphafold2/interactive $
```

```
bash run.sh
```

Interactive: health_ready.sh



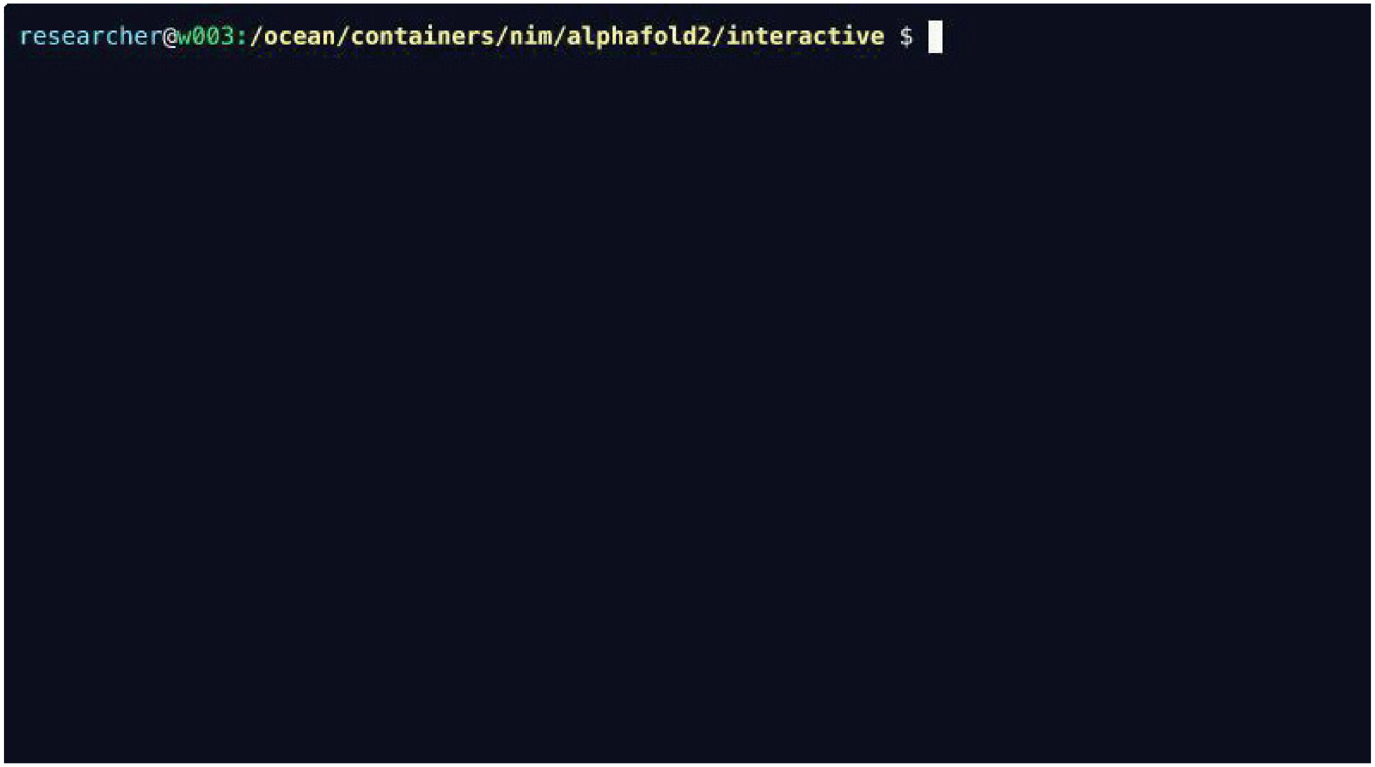
```
#!/bin/env bash

curl http://localhost:8000/v1/health/ready
```



```
researcher@w003:/ocean/containers/nim/alphafold2/interactive $ 
```

```
curl
http://localhost:8000/v1/health/ready
```

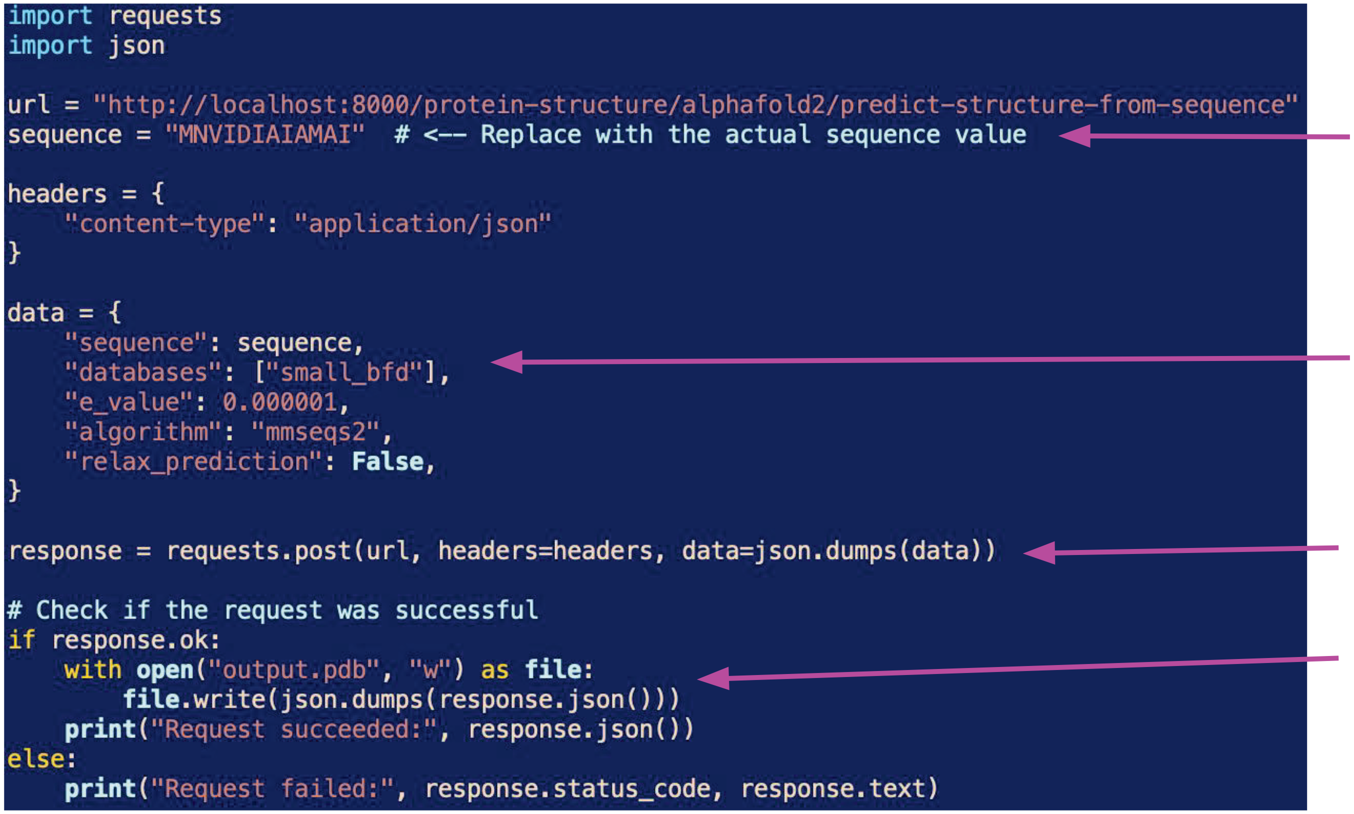
Interactive: nim_client.sh



```
researcher@w003:/ocean/containers/nim/alphafold2/interactive $
```

```
bash nim_client.sh
```

Interactive: nim_client.py



```python
import requests
import json

url = "http://localhost:8000/protein-structure/alphafold2/predict-structure-from-sequence"
sequence = "MNVIDIAIAMAI"  # <-- Replace with the actual sequence value

headers = {
    "content-type": "application/json"
}

data = {
    "sequence": sequence,
    "databases": ["small_bfd"],
    "e_value": 0.000001,
    "algorithm": "mmseqs2",
    "relax_prediction": False,
}

response = requests.post(url, headers=headers, data=json.dumps(data))

# Check if the request was successful
if response.ok:
    with open("output.pdb", "w") as file:
        file.write(json.dumps(response.json()))
    print("Request succeeded:", response.json())
else:
    print("Request failed:", response.status_code, response.text)
```

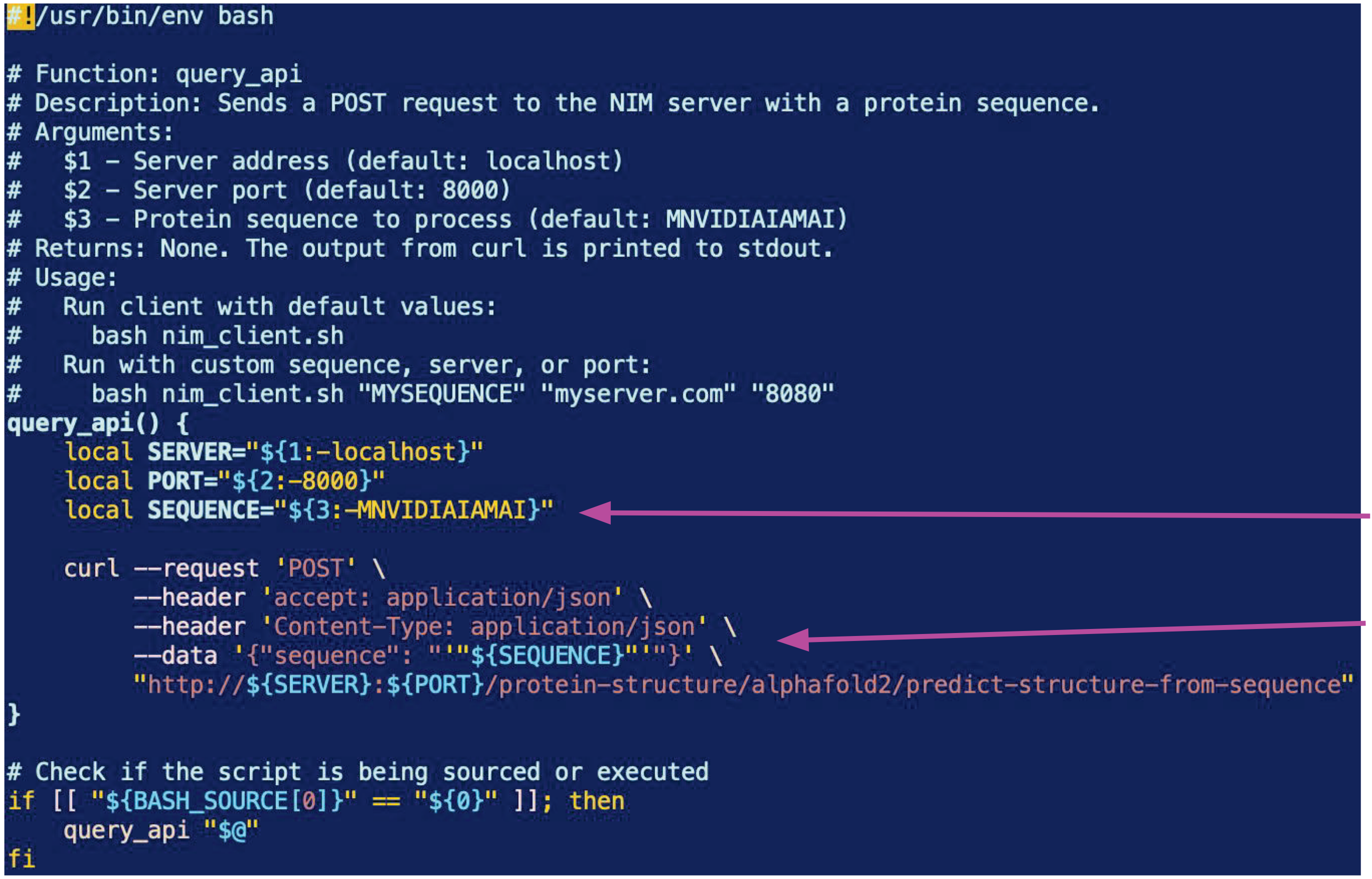
Specify sequence to process

Set additional parameters

Send request

Write back results

Interactive: nim_client.sh

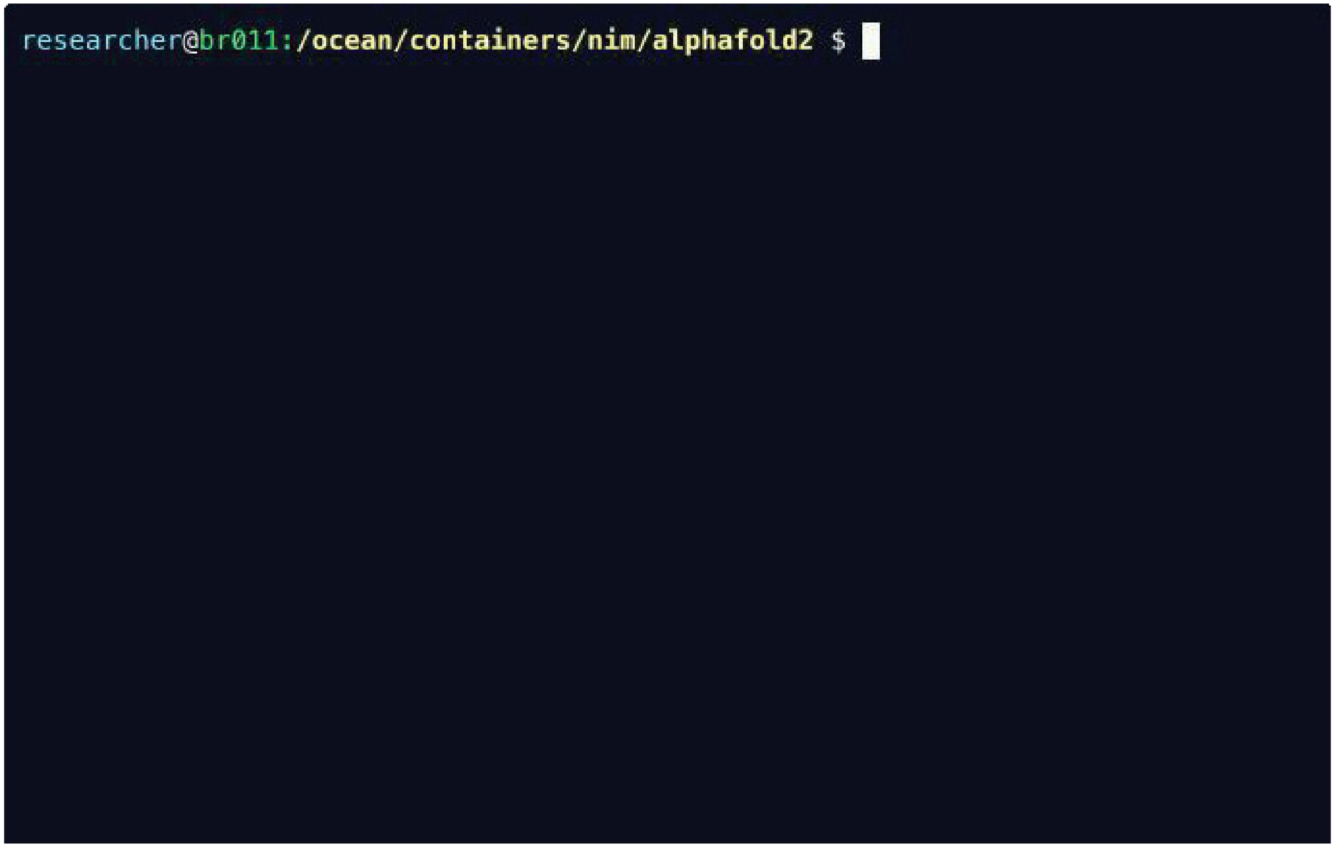


```bash
#!/usr/bin/env bash

# Function: query_api
# Description: Sends a POST request to the NIM server with a protein sequence.
# Arguments:
#   $1 - Server address (default: localhost)
#   $2 - Server port (default: 8000)
#   $3 - Protein sequence to process (default: MNVIDIAIAMAI)
# Returns: None. The output from curl is printed to stdout.
# Usage:
#   Run client with default values:
#     bash nim_client.sh
#   Run with custom sequence, server, or port:
#     bash nim_client.sh "MYSEQUENCE" "myserver.com" "8080"
query_api() {
    local SERVER="${1:-localhost}"
    local PORT="${2:-8000}"
    local SEQUENCE="${3:-MNVIDIAIAMAI}"

    curl --request 'POST' \
         --header 'accept: application/json' \
         --header 'Content-Type: application/json' \
         --data '{"sequence": "'"${SEQUENCE}"'"}' \
         "http://${SERVER}:${PORT}/protein-structure/alphafold2/predict-structure-from-sequence"
}

# Check if the script is being sourced or executed
if [[ "${BASH_SOURCE[0]}" == "${0}" ]]; then
    query_api "$@"
fi
```

Specify sequence to process

Send request

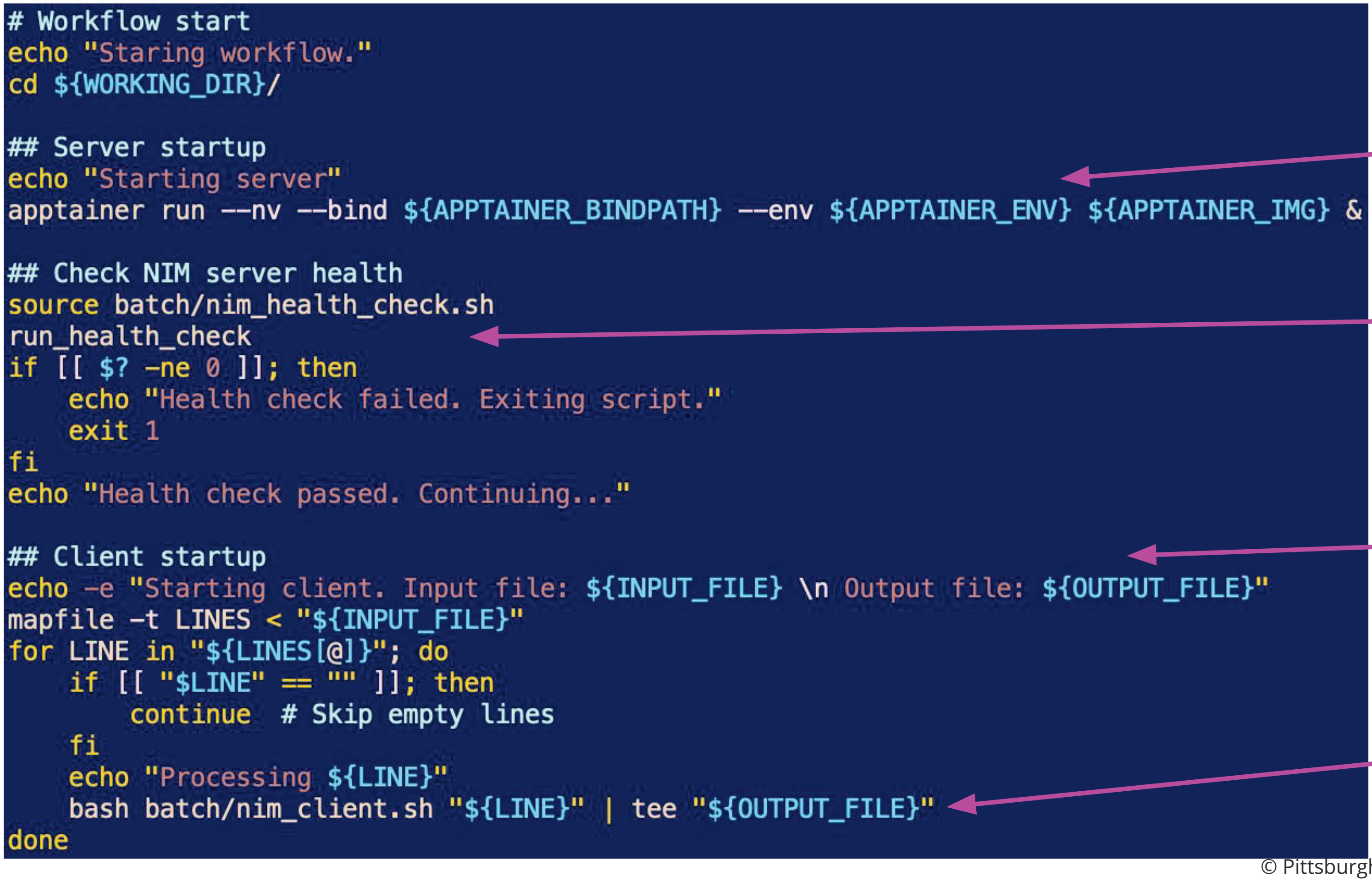
🎬 Batch mode



```
researcher@br011:/ocean/containers/nim/alphafold2 $
```

```
# Edit input_queries.txt
     i.e. MNVIDIAIAMAI

cd /ocean/containers/nim/


cd alphafold2


cd batch


sbatch nim_auto.sbatch
input_file output_file

Example:
sbatch
--output=$HOME/nim_auto.%a.out
nim_auto.sbatch
$HOME/input_queries.txt
$HOME/nim_auto_results.pdb
```

Batch: nim_auto.sbatch



```
# Workflow start
echo "Staring workflow."
cd ${WORKING_DIR}/

## Server startup
echo "Starting server"
apptainer run --nv --bind ${APPTAINER_BINDPATH} --env ${APPTAINER_ENV} ${APPTAINER_IMG} &

## Check NIM server health
source batch/nim_health_check.sh
run_health_check
if [[ $? -ne 0 ]]; then
    echo "Health check failed. Exiting script."
    exit 1
fi
echo "Health check passed. Continuing..."

## Client startup
echo -e "Starting client. Input file: ${INPUT_FILE} \n Output file: ${OUTPUT_FILE}"
mapfile -t LINES < "${INPUT_FILE}"
for LINE in "${LINES[@]}"; do
    if [[ "$LINE" == "" ]]; then
        continue  # Skip empty lines
    fi
    echo "Processing ${LINE}"
    bash batch/nim_client.sh "${LINE}" | tee "${OUTPUT_FILE}"
done
```

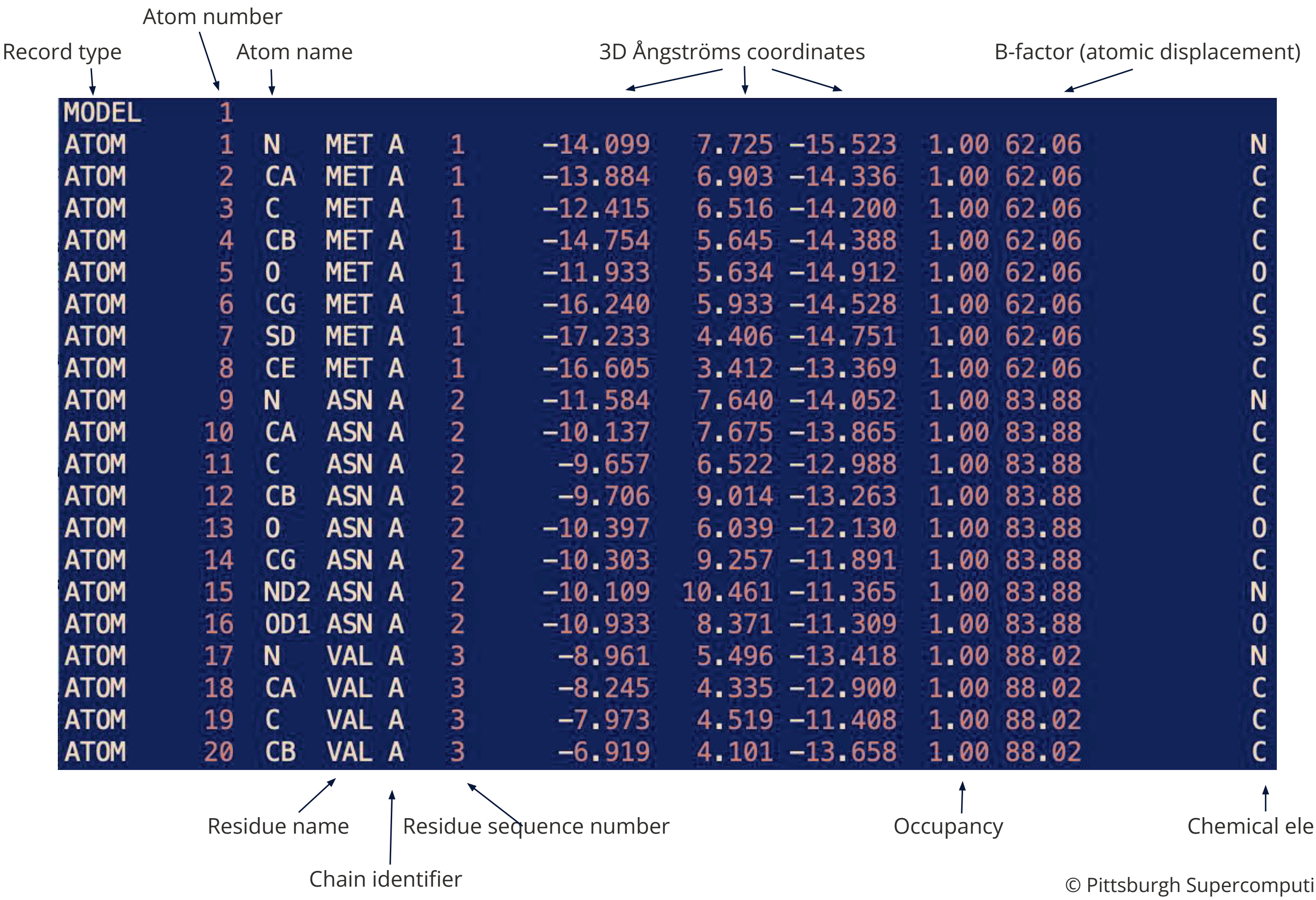
Start up NIM server

Check if the server is up and ready to take queries

Process input file with queries

Send queries to NIM API

Results: output.pdb (Protein Data Bank file)



Atom number

Record type

Atom name

3D Ångströms coordinates

B-factor (atomic displacement)

```
MODEL        1
ATOM         1  N   MET A   1     -14.099   7.725 -15.523  1.00 62.06           N
ATOM         2  CA  MET A   1     -13.884   6.903 -14.336  1.00 62.06           C
ATOM         3  C   MET A   1     -12.415   6.516 -14.200  1.00 62.06           C
ATOM         4  CB  MET A   1     -14.754   5.645 -14.388  1.00 62.06           C
ATOM         5  O   MET A   1     -11.933   5.634 -14.912  1.00 62.06           O
ATOM         6  CG  MET A   1     -16.240   5.933 -14.528  1.00 62.06           C
ATOM         7  SD  MET A   1     -17.233   4.406 -14.751  1.00 62.06           S
ATOM         8  CE  MET A   1     -16.605   3.412 -13.369  1.00 62.06           C
ATOM         9  N   ASN A   2     -11.584   7.640 -14.052  1.00 83.88           N
ATOM        10  CA  ASN A   2     -10.137   7.675 -13.865  1.00 83.88           C
ATOM        11  C   ASN A   2      -9.657   6.522 -12.988  1.00 83.88           C
ATOM        12  CB  ASN A   2      -9.706   9.014 -13.263  1.00 83.88           C
ATOM        13  O   ASN A   2     -10.397   6.039 -12.130  1.00 83.88           O
ATOM        14  CG  ASN A   2     -10.303   9.257 -11.891  1.00 83.88           C
ATOM        15  ND2 ASN A   2     -10.109  10.461 -11.365  1.00 83.88           N
ATOM        16  OD1 ASN A   2     -10.933   8.371 -11.309  1.00 83.88           O
ATOM        17  N   VAL A   3      -8.961   5.496 -13.418  1.00 88.02           N
ATOM        18  CA  VAL A   3      -8.245   4.335 -12.900  1.00 88.02           C
ATOM        19  C   VAL A   3      -7.973   4.519 -11.408  1.00 88.02           C
ATOM        20  CB  VAL A   3      -6.919   4.101 -13.658  1.00 88.02           C
```

Residue name

Residue sequence number

Occupancy

Chemical element

Chain identifier

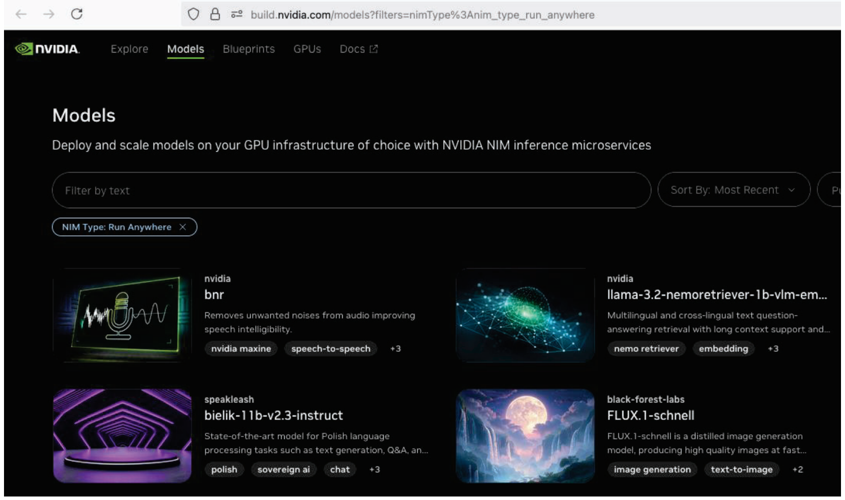
🛠️ Requesting a New NIM

Steps for requesting a new container.

📘 Where to Start

Step 1: Find Your NIM

- Already available ones? → [Bridges-2 User Guide]
- Go to build.nvidia.com → Models → [NIM Type: Run Anywhere]
    - ○ Find the one that fits your task



build.nvidia.com/models?filters=nimType%3Anim_type_run_anywhere

Models

Deploy and scale models on your GPU infrastructure of choice with NVIDIA NIM inference microservices

nvidia
bnr
Removes unwanted noises from audio improving speech intelligibility.
nvidia maxine · speech-to-speech · +3

nvidia
llama-3.2-nemoretriever-1b-vlm-em...
Multilingual and cross-lingual text question-answering retrieval with long context support and...
nemo retriever · embedding · +3

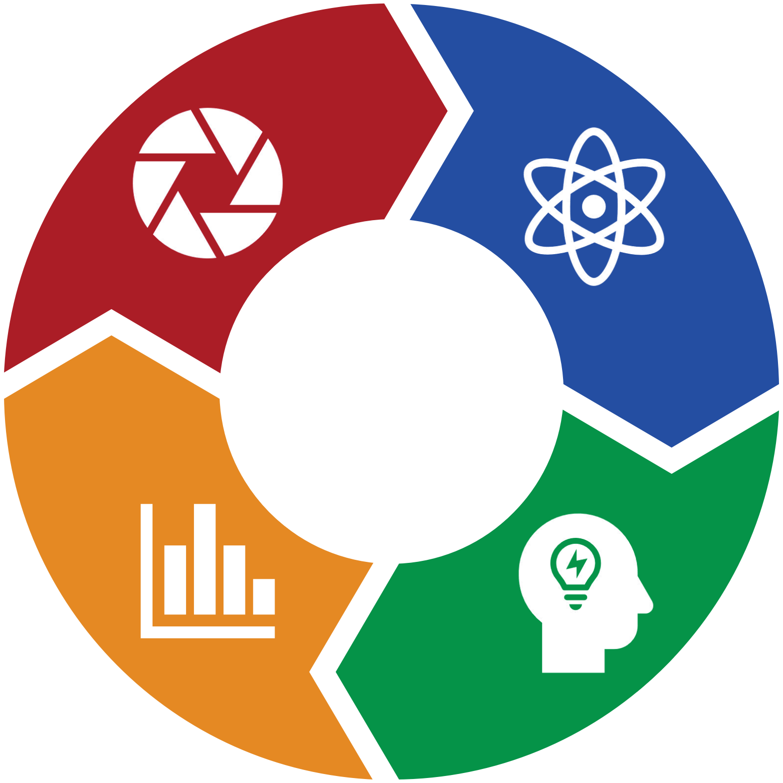
speakleash
bielik-11b-v2.3-instruct
State-of-the-art model for Polish language processing tasks such as text generation, Q&A, an...
polish · sovereign ai · chat · +3

black-forest-labs
FLUX.1-schnell
FLUX.1-schnell is a distilled image generation model, producing high quality images at fast...
image generation · text-to-image · +2

Step 2: Request New Ones (If Needed)

- Fill this form: psc.edu → [NIM Container Request Form]
- We handle the setup.

What Happens Behind the Scenes (FYI)



STAGE 1: Pull

We fetch & test the container

STAGE 2: Configure

Prepare usage scripts

STAGE 3: Document

Document notes, best practices.

STAGE 4: Publish

Add to the Bridges-2 catalog, available under: **`/ocean/containers/nim/<container_name>/`**

📎 Recap

Pick Your Workflow:

- 🧾 **Batch:** quick, easy, no code
- 🖥️ **Interactive:** flexible, real-time

Need a NIM?

- **See what's ready:** Bridges-2 User Guide → Containers -> NIM Containers
- **Browse:** build.nvidia.com → Models → NIM Type: Run Anywhere
  - Find the one that fits your task
- **Request:** Bridges-2 User Guide → Containers → NIM Containers → Request a NIM Container on Bridges-2 Form

🤔 Q&A

Happy to help you get started—ask away.

Where to Get Help: PSC Support (Email `help@psc.edu`)

Consider applying to participate in the PSC/CMU/Pitt Hackathon in partnership with NVIDIA:

- PSC/CMU/Pitt Open Hackathon | Open Hackathons - **Deadline is June 25**.