Argonne NATIONAL LABORATORY

Coupling streaming AI and HPC ensembles to achieve 100-1000× faster biomolecular simulations

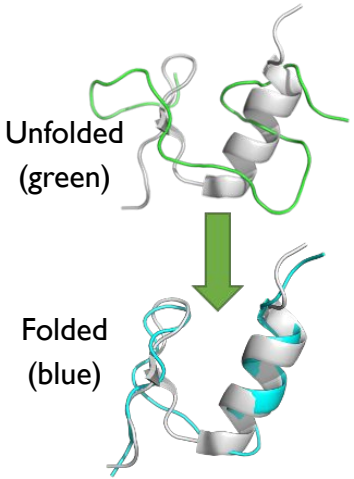
Alexander Brace[1,3*], Igor Yakushin[1*], Heng Ma[1*], Anda Trifan[1,4], Todd Munson[2✢], Ian Foster[1,3✢], Arvind Ramanathan[1✢], Hyungro Lee[5*], Matteo Turilli[5,6], Shantenu Jha[5,6✢]

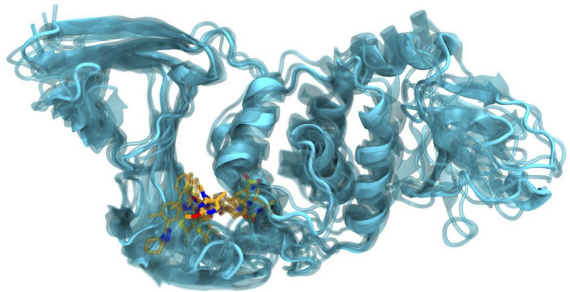
[1]Data Science and Learning Division, Argonne National Laboratory, [2]Mathematics and Computer Science Division, Argonne National Laboratory, [3]University of Chicago, [4]University of Illinois Urbana-Champaign, [5]Rutgers University, [6]Computational Science Initiative, Brookhaven National Laboratory; ✢Senior authors *First authors

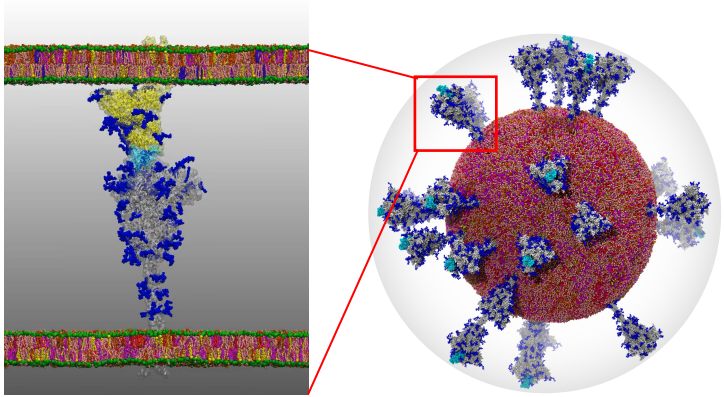
U.S. DEPARTMENT OF ENERGY Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

ParslFest, Oct 19, 2023

Science motivation: Studying complex biological systems and processes is important







O(10M atoms) O(1B atoms)

Protein folding provides insights into how their 3D structure enables function e.g. FSD-EY ($\beta\beta\alpha$)

Protein ligand complexes evaluate drug binding to discover novel inhibitors e.g. Papain-like protease (PLPro)

Many-Atom Multiscale Systems uncover large-scale dynamics necessary to understand systems such as SARS-CoV-2

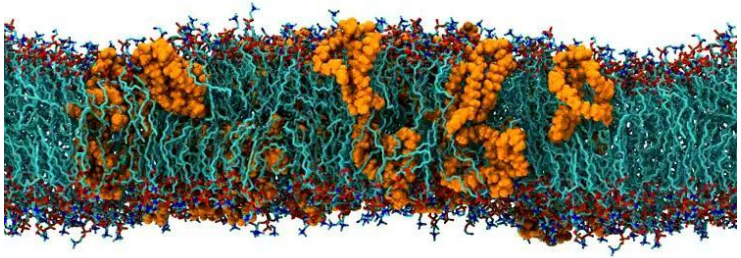
Casalino, L. et al, **AI-driven multiscale simulations illuminate mechanisms of SARS-CoV-2 spike dynamics,** https://doi.org/10.1177/10943420211006452 (SC'20 Gordon Bell Award for HPC in COVID-19 Research)

What is molecular dynamics and why do we do it?

- Molecular dynamics (MD) can act as a **computational microscope** which can reveal biomolecular detail at greater spatial and temporal scales than is possible with experiment

- Integrates Newton's second law of motion to advance each atom in the system forward in time along a potential energy gradient

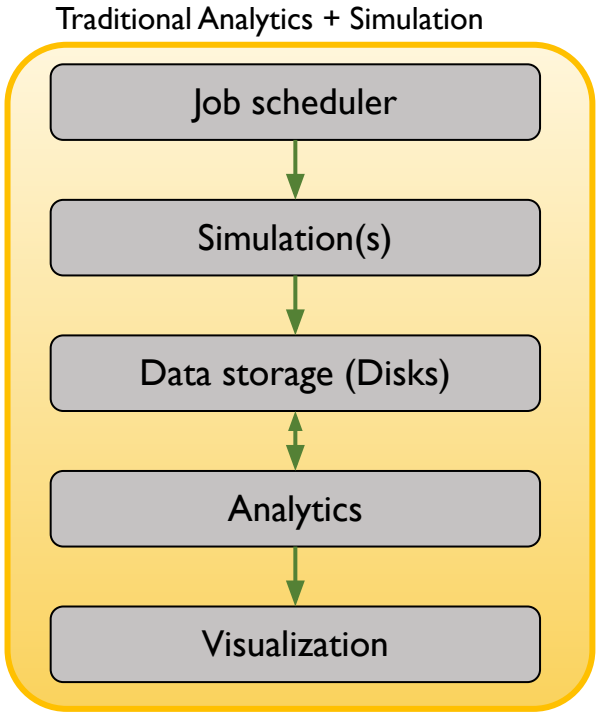
$$\vec{F}_i = ma = m_i \frac{d^2\vec{r}_i}{dt^2} = -\vec{\nabla}U(\vec{R})$$

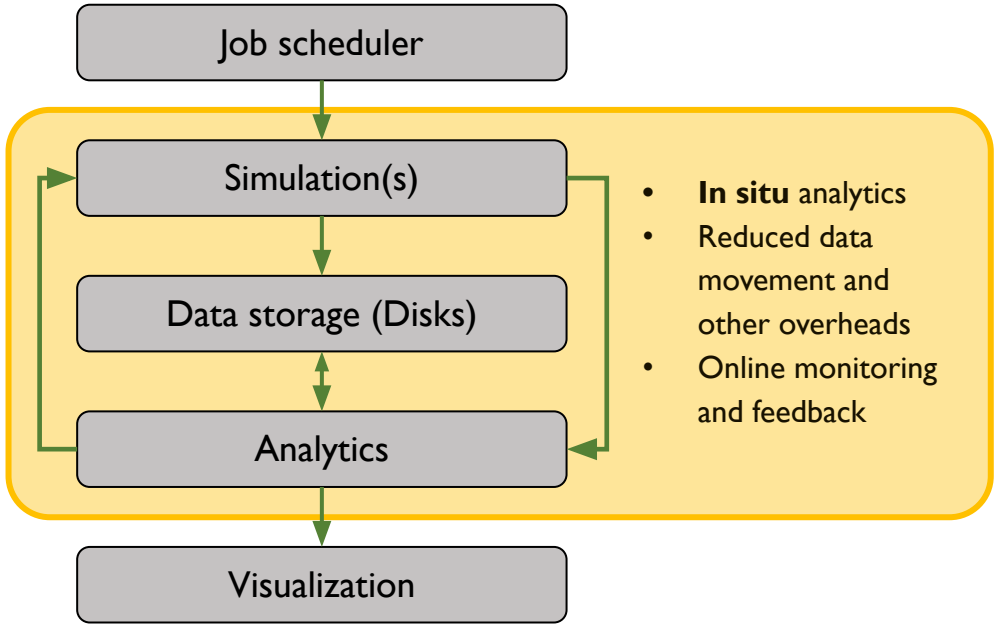


Membrane simulation

R. O. Dror et al. **Biomolecular simulation: A computational microscope for molecular biology**

Multiscale simulations impose new requirements on emerging hardware/software



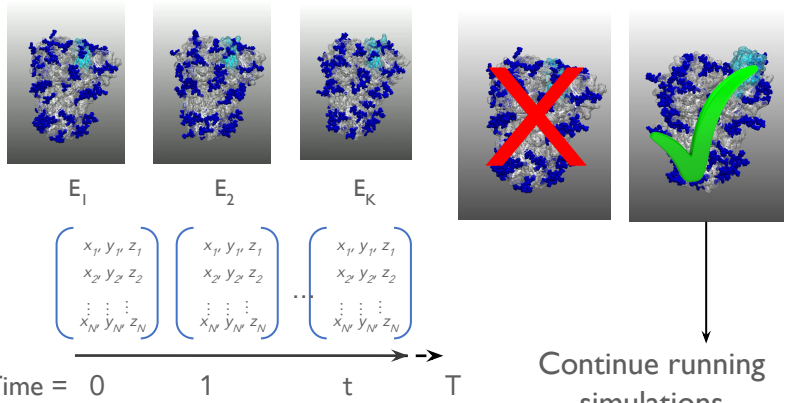


- Large simulations generate **> O(100 TB)** of data
- Humanly impossible to peek into "biologically" interesting events!
- Traditional method is unsustainable at exascale

AI-enabled MD simulations with DeepDriveMD

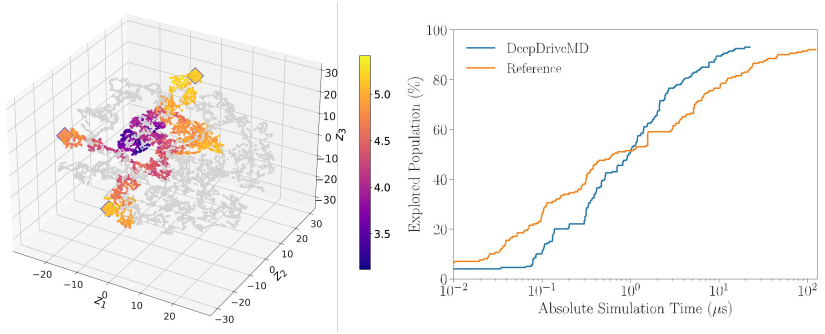
Coordinates, contact maps, other features

Ensemble MD simulations



Continue running simulations

Deep Learning/ Artificial Intelligence



Build physically interpretable embeddings

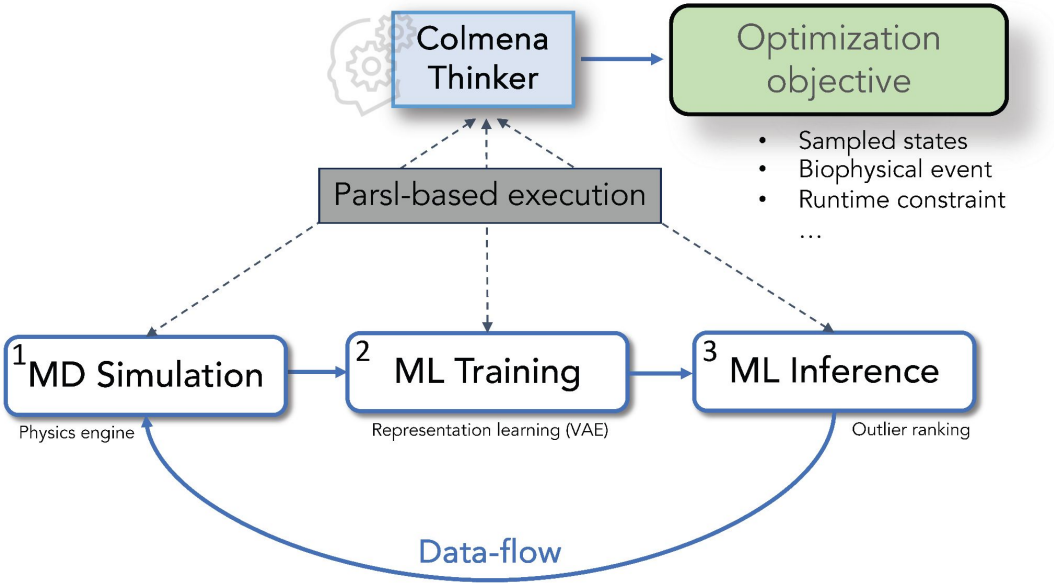
Track states that are sampled more often

"Interesting conformations", population sampled

Learning Everywhere

- Jha & Fox. In "Visionary Track", 15th International Conference eScience (2019), San Diego, California

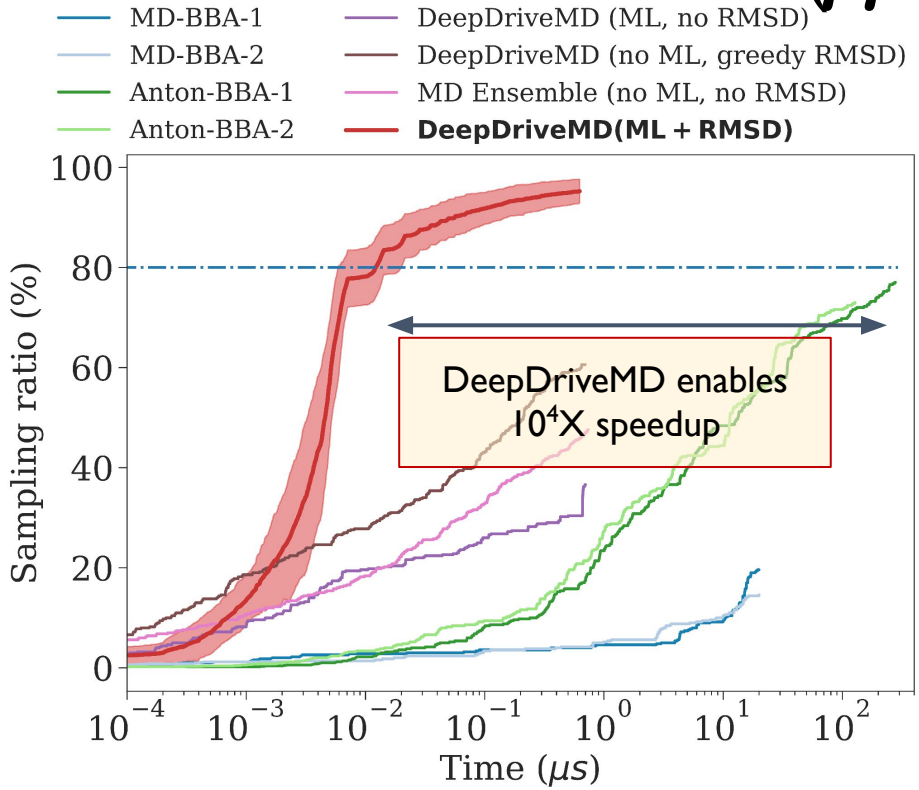
DeepDriveMD with Colmena



- **Execution:** Each task is executed as an independent function call via Parsl, as orchestrated by the Colmena "Thinker".
- **Parallelism:** Decoupled design of components allows for high degrees of parallelism (e.g., 100s of parallel simulation tasks).
- **Communication:** Each component writes outputs to disk (e.g., simulation data, model weights, etc) and a path to the data is returned via Proxystore. However, other communication protocols can be used without change to the framework.

Ward, L. et al., **Colmena: Scalable Machine-Learning-Based Steering of Ensemble Simulations for High Performance Computing**., https://arxiv.org/abs/2110.02827

DeepDriveMD enables 10^4X acceleration of sampling effectiveness for FSD-EY ($\beta\beta\alpha$) folding



- Embedding states into the VAE latent space and clustering with k-means keeps a constant definition of the number of states sampled enabling fair comparison between simulations
- The ML + RMSD strategy reaches 80% sampling more than 1000X faster than Anton-1 simulations

Note: Uncertainty from 10 trials is shown in light red

Brace, A. et al., **Achieving 100X faster simulations of complex biological phenomena by coupling ML to HPC ensembles**., https://arxiv.org/abs/2104.04797

Modular Simulate-Train-Infer Framework

```python
class DeepDriveMDWorkflow(BaseThinker):  # type: ignore[misc]

    @abstractmethod
    def simulate(self) -> None:
        """Start a simulation task.

        Must call :meth:`submit_task` with ``topic='simulation'``"""
        ...

    @abstractmethod
    def train(self) -> None:
        """Start a training task.

        Must call :meth:`submit_task` with ``topic='train'``"""
        ...

    @abstractmethod
    def inference(self) -> None:
        """Start an inference task

        Must call a :meth:`submit_task` with ``topic='infer'``"""
        ...
```

```python
    @abstractmethod
    def handle_simulation_output(self, output: Any) -> None:
        """Stores a simulation output in the training set and define new inference tasks

        Should call ``self.run_training.set()`` and/or ``self.run_inference.set()``

        Parameters
        ----------
        output:
            Output to be processed
        """
        ...

    @abstractmethod
    def handle_train_output(self, output: Any) -> None:
        """Use the output from a training run to update the model"""
        ...

    @abstractmethod
    def handle_inference_output(self, output: Any) -> None:
        """Use the output from an inference run to update the list of available simulations"""
        ...
```

Interface: An abstraction layer over the particular simulation / training / inference tasks allows users to customize DeepDriveMD to any simulation engine or ML method

Pydantic Parsl configuration

```python
class BaseComputeSettings(BaseSettings, ABC):
    """Compute settings (HPC platform, number of GPUs, etc)."""

    name: Literal[""] = ""
    """Name of the platform to use."""

    @abstractmethod
    def config_factory(self, run_dir: PathLike) -> Config:
        """Create a new Parsl configuration.

        Parameters
        ----------
        run_dir : PathLike
            Path to store monitoring DB and parsl logs.

        Returns
        -------
        Config
            Parsl configuration.
        """
        ...
```

- The correct compute settings are automatically determined via Pydantic type checking by specifying the "name" field in YAML along with conditional args

Code: https://github.com/ramanathanlab/deepdrivemd/tree/main

```python
class WorkstationSettings(BaseComputeSettings):
    name: Literal["workstation"] = "workstation"  # type: ignore[assignme
    """Name of the platform."""
    available_accelerators: Union[int, Sequence[str]] = 8
    """Number of GPU accelerators to use."""
    worker_port_range: Tuple[int, int] = (10000, 20000)
    """Port range."""
    retries: int = 1
    label: str = "htex"

    def config_factory(self, run_dir: PathLike) -> Config:
        return Config(
            run_dir=str(run_dir),
            retries=self.retries,
            executors=[
                HighThroughputExecutor(
                    address="localhost",
                    label=self.label,
                    cpu_affinity="block",
                    available_accelerators=self.available_accelerators,
                    worker_port_range=self.worker_port_range,
                    provider=LocalProvider(init_blocks=1, max_blocks=1),
                ),
            ],
        )
```

```yaml
compute_settings:
  # Specify we want the workstation parsl configuration
  name: workstation
  # Identify which GPUs to assign tasks to. It's generally recommended to first check
  # nvidia-smi to see which GPUs are available. The numbers below are analogous to
  # setting CUDA_VISIBLE_DEVICES=0,1,2,3
  available_accelerators: ["0", "1", "2", "3"]
```

Acknowledgements

Logan Ward & Kyle Chard for supporting the Parsl/Colmena-based implementation of DeepDriveMD

Rick Stevens (Argonne/ University of Chicago)

Rommie Amaro (University of California San Diego)

John Stone (University of Illinois Urbana-Champaign)

Lillian Chong (University of Pittsburgh)

Tom Gibbs (NVIDIA)

~300 collaborators on COVID-19 research

Funding:

-- DOE ASCR Co-design of AI Approaches for Multi-modal datasets

-- Exascale Computing Project (ECP): Cancer Deep Learning Environment (CANDLE), ECP Co-design for Online Data Analysis and Reduction CODAR, ExaWorks

-- Department of Energy's Advanced Scientific Computing Research Program

-- DOE National Virtual Biotechnology Laboratory

Computing:

-- Argonne/Oak Ridge Leadership Computing Facilities

-- Livermore Computing at the Lawrence Livermore National Laboratory

-- HPC Consortium for COVID-19 research

THANK YOU!!

QUESTIONS/ COMMENTS:

RAMANATHANA@ANL.GOV