Structure-Based Virtual Screening and Data Analytics on Summit for COVID-19

Jens Glaser, Computational Scientist

Oak Ridge Leadership Computing Facility

Oak Ridge National Laboratory

2021 OLCF User Meeting, June 22nd

Acknowledgements

NCCS/CSM: Josh Vermaas, David Rogers, Swen Boehm, Matthew Baker, Oscar Hernandez, Ben Hernandez, Suhas Somnath, Jason Kincl, Ketan Maheshwari, Jess Woods (ORISE), Dustin Leverman, Don Maxwell, Arjun Shankar, Ryan Prout, Spencer Ward, Jeff Nichols, Jack Wells (ORNL LDRD SEED Committee), Gina Tourassi, Darren Hsu, Andrew Blanchard, Debsindhu Bhowmik, Dmytro Bykov, Dayle Smith, Dale Stansberry, Stephan Irle

CMB/BSD/JIBS: **Ada Sedova**, Jeremy Smith, Jerry Parks, Stephanie Galanie, Rupesh Agarwal, Marti Head, Daniel Kneller, Andrii Kovalevskyi, Omar Demerdash, Stan Martin, Bob Hettich, Micholas Smith, Julie Mitchell, Russ Davidson, Audrey Labbe

NVIDIA: **Jeff Larkin**, **Scott LeGrand,** John Kirkham, Josh Patterson, John Eaton, Duncan Poole, Jon Lefman, Geetika Gupta

Scripps Research: **Andreas Tillack**, Diogo Santos-Martins, Stefano Forli

BlazingSQL: Rodrigo Aramburu, Felipe Aramburu, William Malpica, Shannon Smith

Google: Jamey Kinney, Usman Qureshi, Miles Euell

JubileeDev: **Aaron Scheinberg**

Funding: Research sponsored by the emergency funding to NCCS through the ASCR / DOE CARES act, and Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL). This research used resources of the Oak Ridge Leadership Computing Facility at ORNL and Google Cloud, through the White House Covid HPC Consortium. ORNL is managed by UT-Battelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

'20 COVID-19 GB finalist team

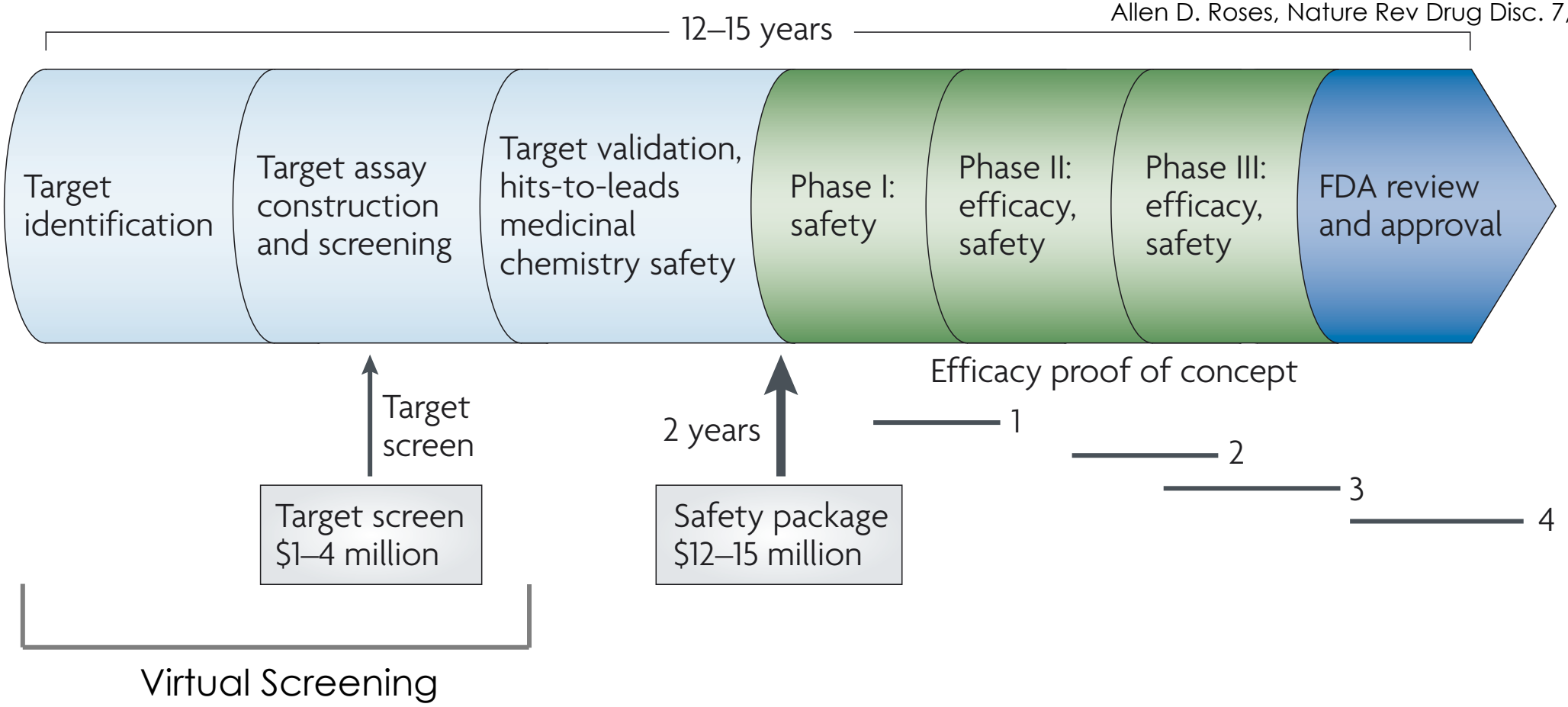
COVID-19 Antivirals

- FDA-approved: Remdesivir (polymerase inhibitor)
- In clinical trials:
  - Molnupiravir (*polymerase inhibitor*, phase II/III)
  - PF-07321332 (*protease inhibitor*, phase-I)
  - PF-07304814 (*protease inhibitor*, phase-I)
  - Monoclonal antibodies (*entry inhibitors*, phase I/II)
- $3.2B US federal funding for a COVID-19 antiviral by end of 2021

https://covdb.stanford.edu/page/covid-reviev

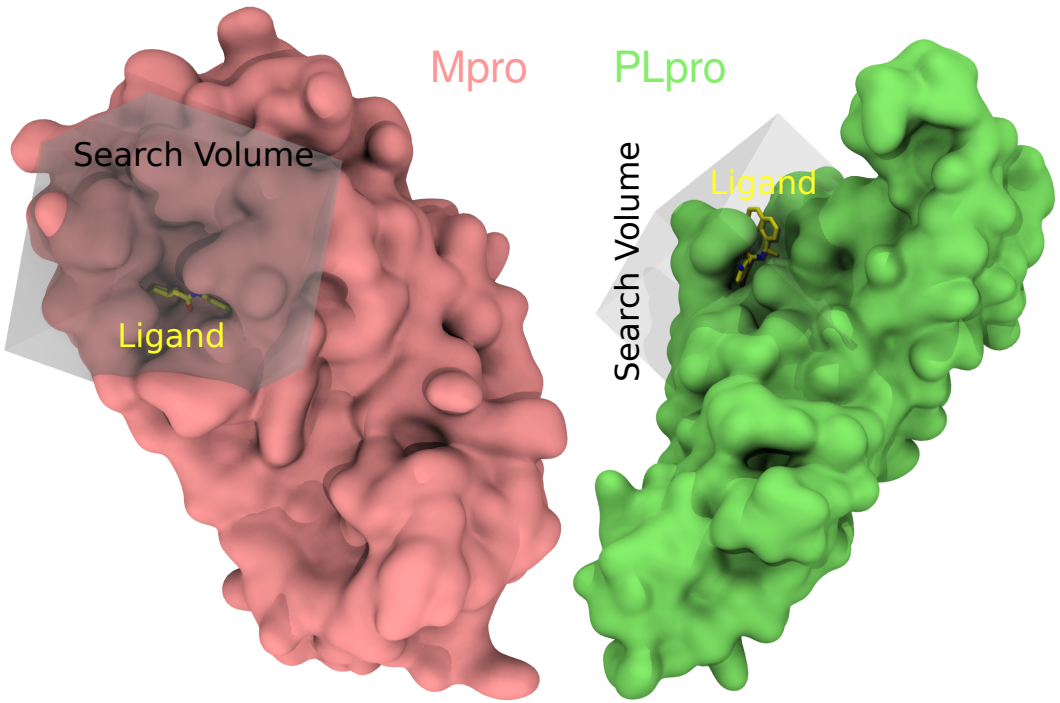
Typical Drug Discovery Timeline

Allen D. Roses, Nature Rev Drug Disc. 7, 807-817 (2008)



Can we shorten the molecule screening phase to a few months using HPC with GPUs?

Structure-based Drug Discovery

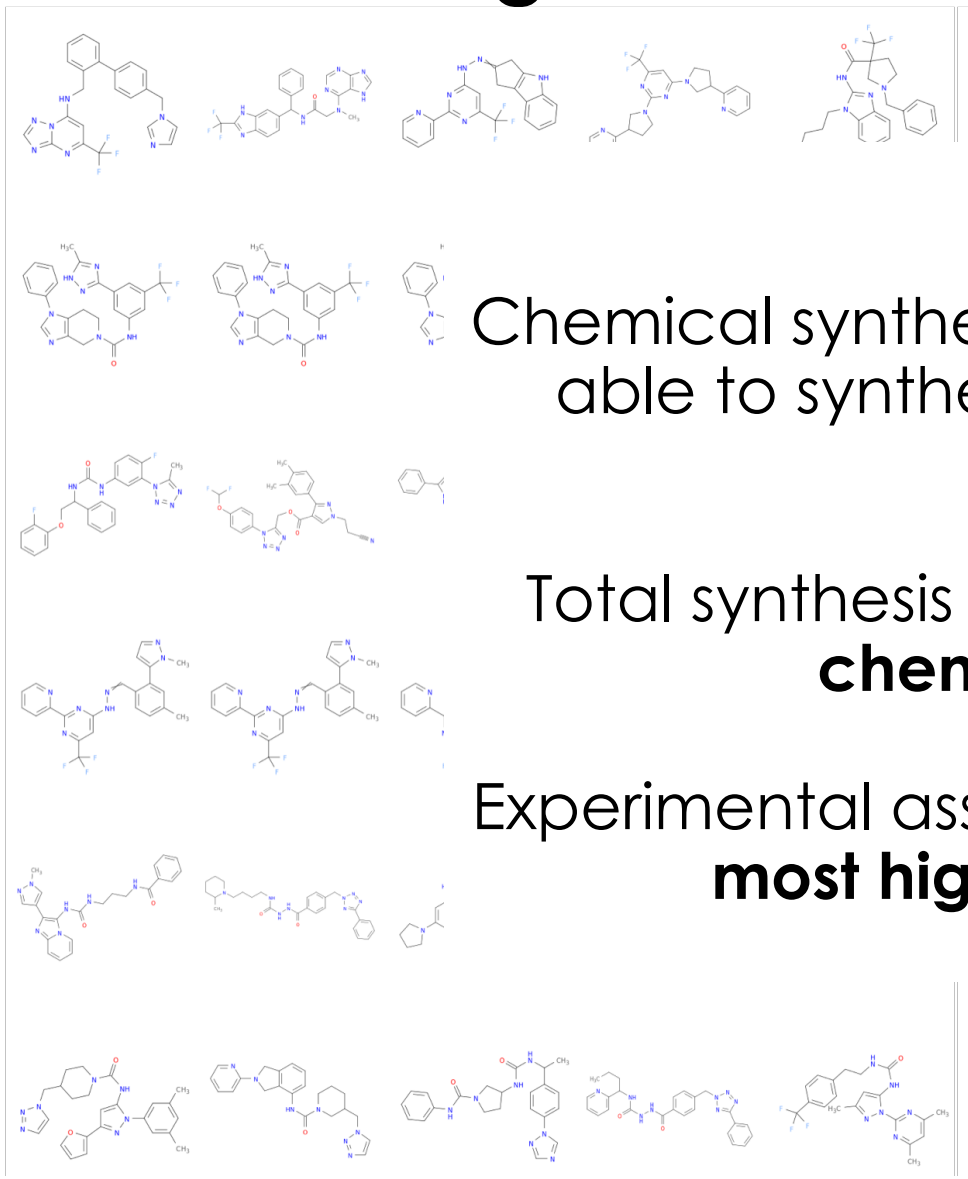


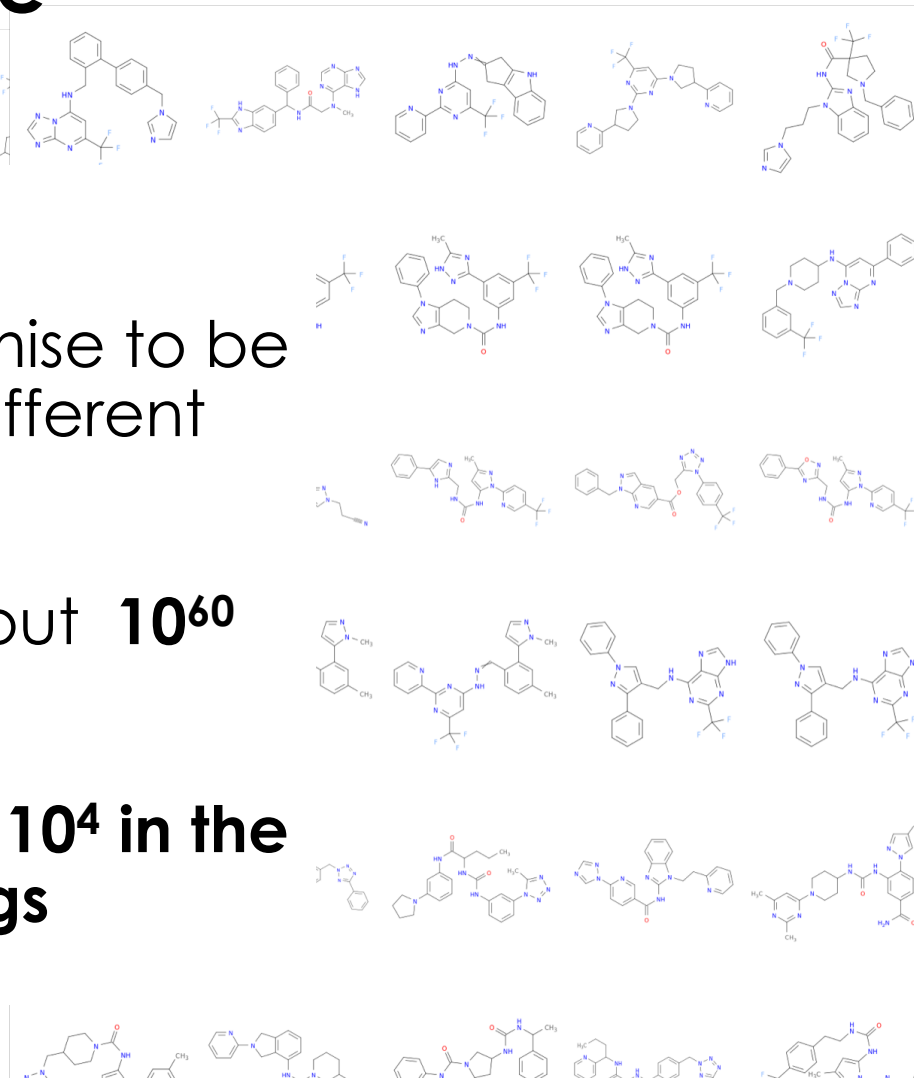
Two of the SARS-CoV-2 protein targets of close to 30 total proteins

- **Structure-based drug discovery** uses three-dimensional models of small molecules binding to protein "receptors"
  - For COVID-19 many groups are targeting the viral proteins in order to find molecules that can inhibit viral entry and replication
  - These small molecule compounds, or "ligands" could be used to develop potential drugs
  - By binding to the receptor's binding site, a small molecule can inhibit the protein's action
- Molecular docking is an optimization calculation within a biomolecular simulation

Vermaas, Josh Vincent, Ada Sedova, Matthew Baker, Swen Boehm, David Rogers, Jeff Larkin, Jens Glaser, Micholas Smith, Oscar Hernandez, and Jeremy Smith. "Supercomputing Pipelines Search for Therapeutics Against COVID-19." *Computing in Science & Engineering* (2020). https://doi.org/10.1109/MCSE.2020.3036540

Searching the Vast Chemical Space





Chemical synthesis companies promise to be able to synthesize over 1 billion different compounds

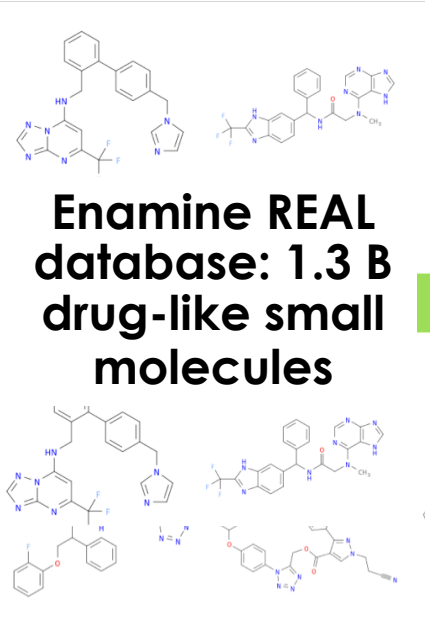
Total synthesis space contains about **10^{60} chemical compounds***

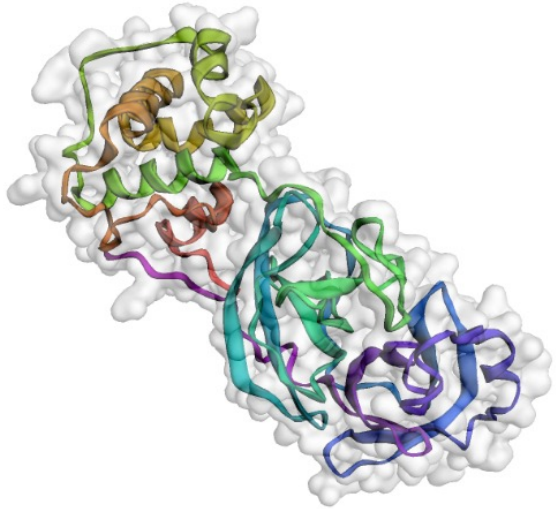
Experimental assays can test about **10^{4} in the most high-throughput settings**

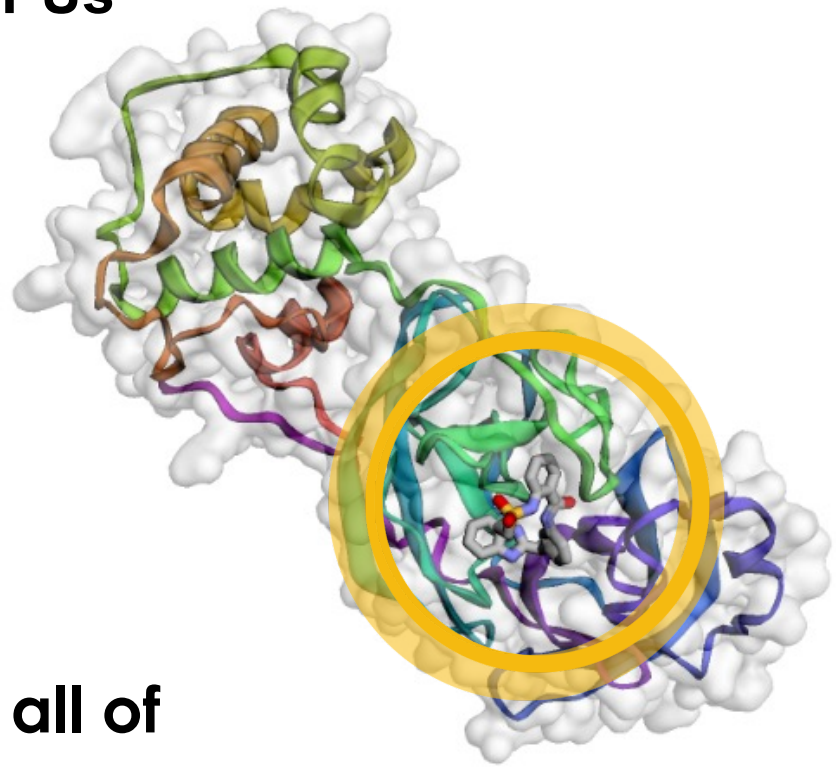
*Virshup et al. J. Am. Chem. Soc. 2013

Docking 1.3 billion compounds to SARS-CoV-2 protein in under 24 hours

using all of Summit's GPUs and CPUs

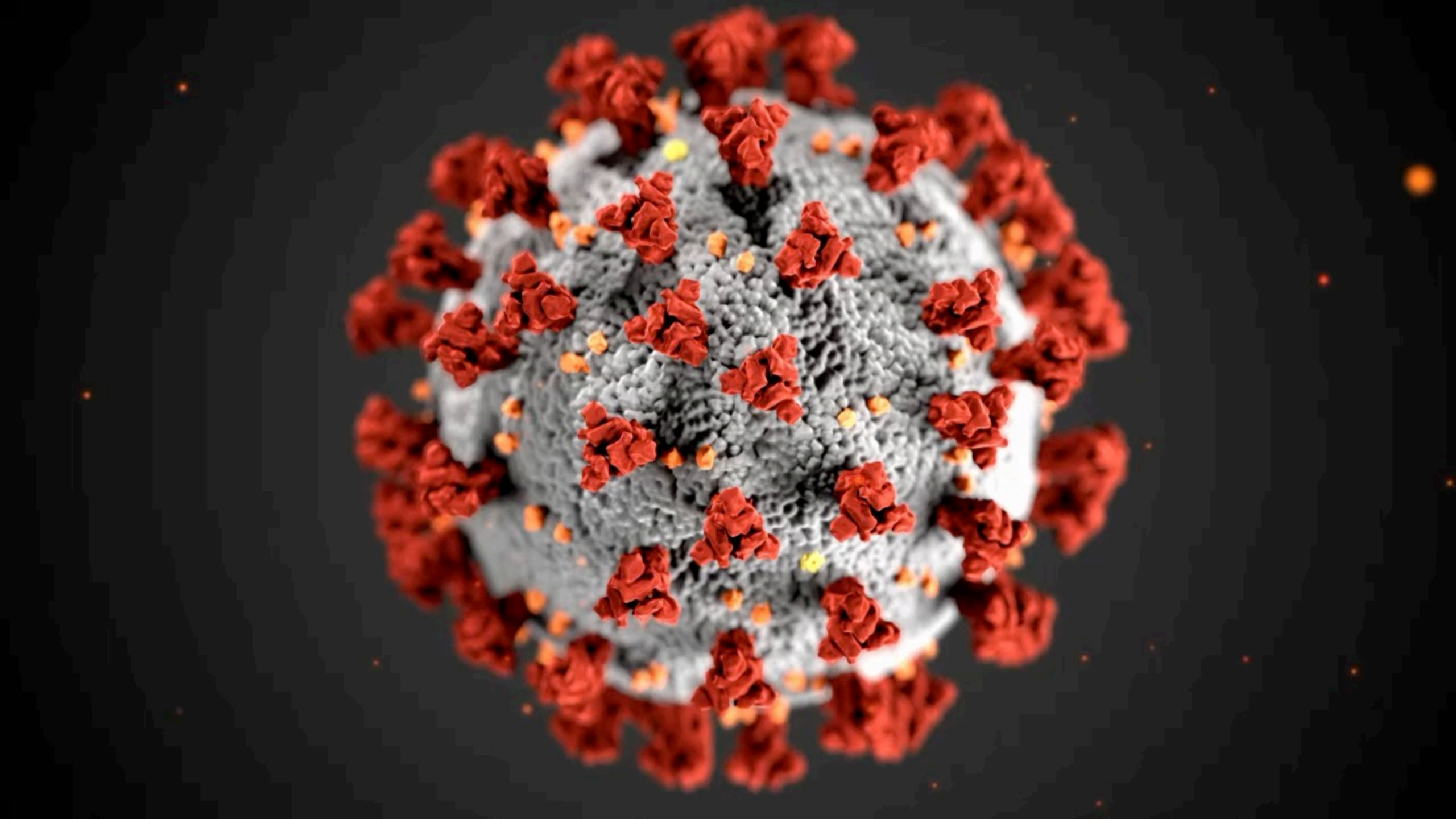






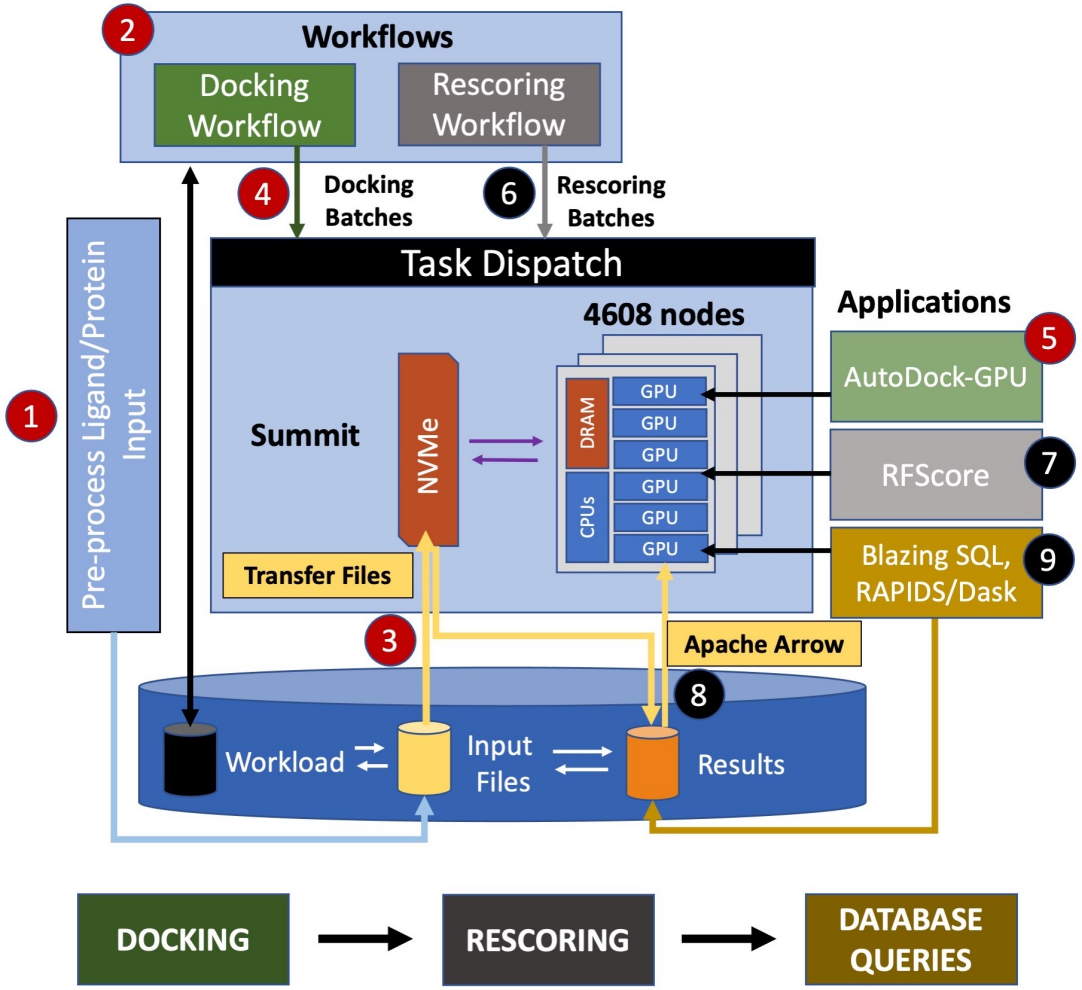
Drug molecule binds to SARS-CoV-2 protein as simulated on Summit with Autodock-GPU

- **Docked 2.6 billion compounds in 2 days using all of Summit (1.3B/day)**
- Full-accuracy molecular docking with **optimization of ligand internal coordinates** and generating 20 poses per docking



Accelerating the End-to-end Pipeline >50×

1. **High-throughput AutoDock-GPU on Summit: Docking Billions of Compounds at Scale for COVID-19 Drug Discovery**
2. Accelerated Kernels for Machine Learning Feature Calculations: Better Predictions via ML-based Rescoring
3. Data Analytics on Massive Outputs Within a GPU-accelerated Virtual Laboratory



Glaser, J., Vermaas, J.V., Rogers, D.M., Larkin, J., LeGrand, S., Boehm, S., Baker, M.B., Scheinberg, A., Tillack, A.F., Thavappiragasam, M. and Sedova, A., 2021. High-throughput virtual laboratory for drug discovery using massive datasets. *The International Journal of High Performance Computing Applications*, https://doi.org/10.1177/10943420211001565

Tackling Immediate Scaling Challenges for High-throughput Docking Calculations on Summit

- There were multiple codes to choose from at the start of the pandemic
  - *AutoDock-GPU from Scripps Research was in development*
- Most docking codes only use the CPU
  - *97% of the FLOPs on Summit are on the GPUs*
- In original code, each small molecule/protein pair runs a distinct executable
  - *Each instance reads and reloads the protein file even if the same one is used repeatedly*

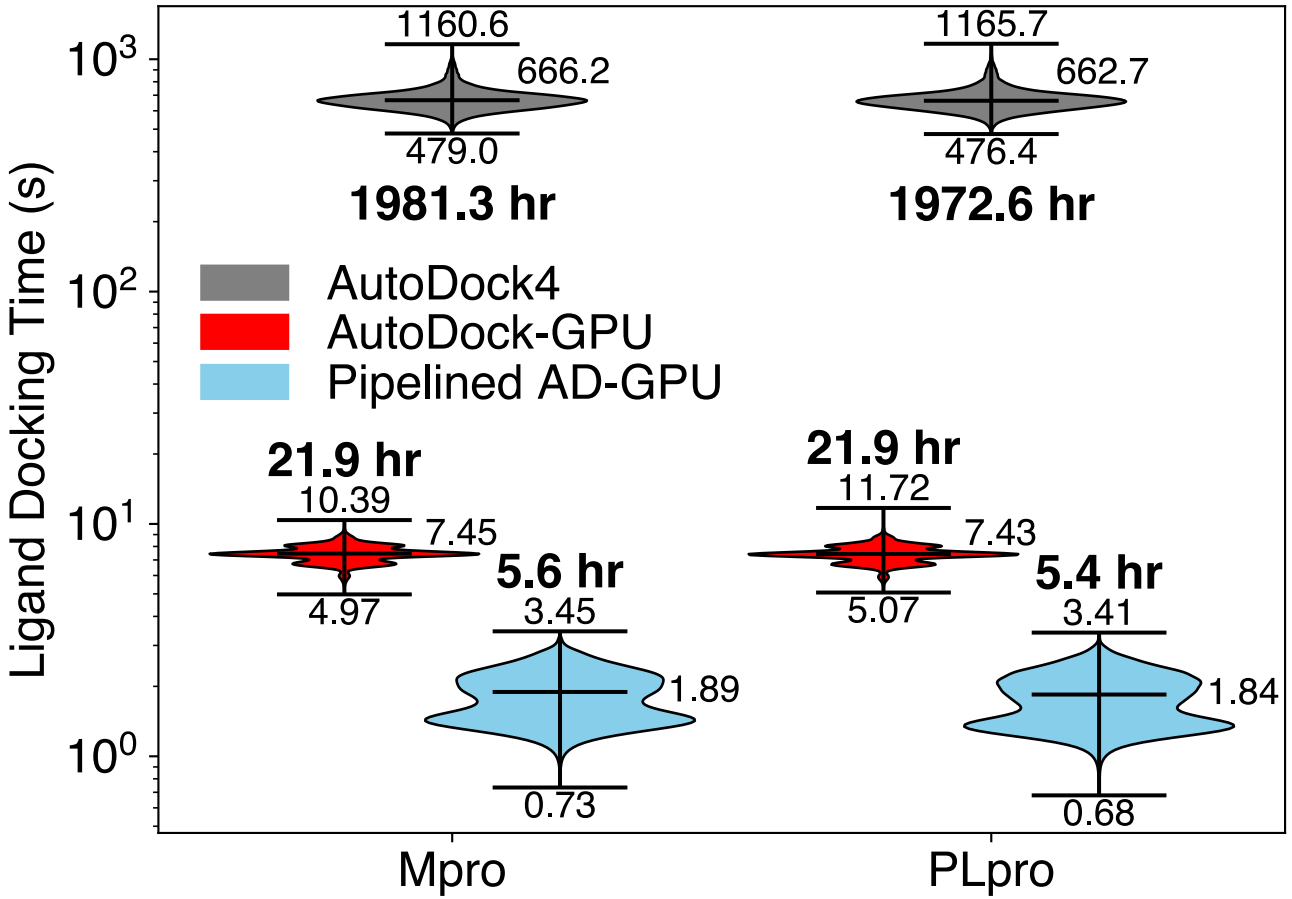
Program	Reference	License	GPU	2020 Citations
AutoDock Vina	Trott and Olson (2010)	Apache		2330
AutoDock4	Morris et al. (2009)	GNU		1670
GLIDE	Friesner et al. (2004)	Commercial		569
DOCK6	Allen et al. (2015)	Academic		59
rDock	Ruiz-Carmona et al. (2014)	GNU		53
FRED	McGann (2011)	Commercial		43
DOCK3	Coleman et al. (2013)	Academic		17
PLANTS	Korb et al. (2006)	Academic	✔Korb et al. (2011)	16
QuickVina-W	Hassan et al. (2017)	Apache		15
QuickVina 2	Alhossary et al. (2015)	Apache		12
BUDE	McIntosh-Smith et al. (2015)	Unavailable	✔	8
GeauxDock	Fang et al. (2016)	Academic	✔	5
AutoDock-GPU	Santos-Martins et al. (2019b)	GNU	✔	4
GOLD		Commercial		
LeDock		Commercial		
MOE-dock		Commercial		

AutoDock-GPU

- **A new GPU version from Scripps based on the widely used AutoDock4 program**
- **Well supported**
- **Open Source**

Production High-throughput Version: Summit

- GPU version gains an average of **350× speedup over CPU serial version** for our test set (Enamine Diversity Set, 10K different compounds)
  - Individual calculations take seconds
- File loading and CUDA setup are a significant portion of the runtime
  - Reusing CUDA context, data and files between ligands substantially accelerates the runs
- Average of 50× speedup per Summit node vs. CPU version run on all 42 cores



LeGrand, S., Scheinberg, A., Tillack, A.F., Thavappiragasam, M., Vermaas, J.V., Agarwal, R., Larkin, J., Poole, D., Santos-Martins, D., Solis-Vasquez, L. and Koch, A., 2020, September. GPU-Accelerated Drug Discovery with Docking on the Summit Supercomputer: Porting, Optimization, and Application to COVID-19 Research. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* (pp. 1-10). https://doi.org/10.1145/3388440.3412472

Deploying Autodock-GPU on Summit at Scale

Fireworks

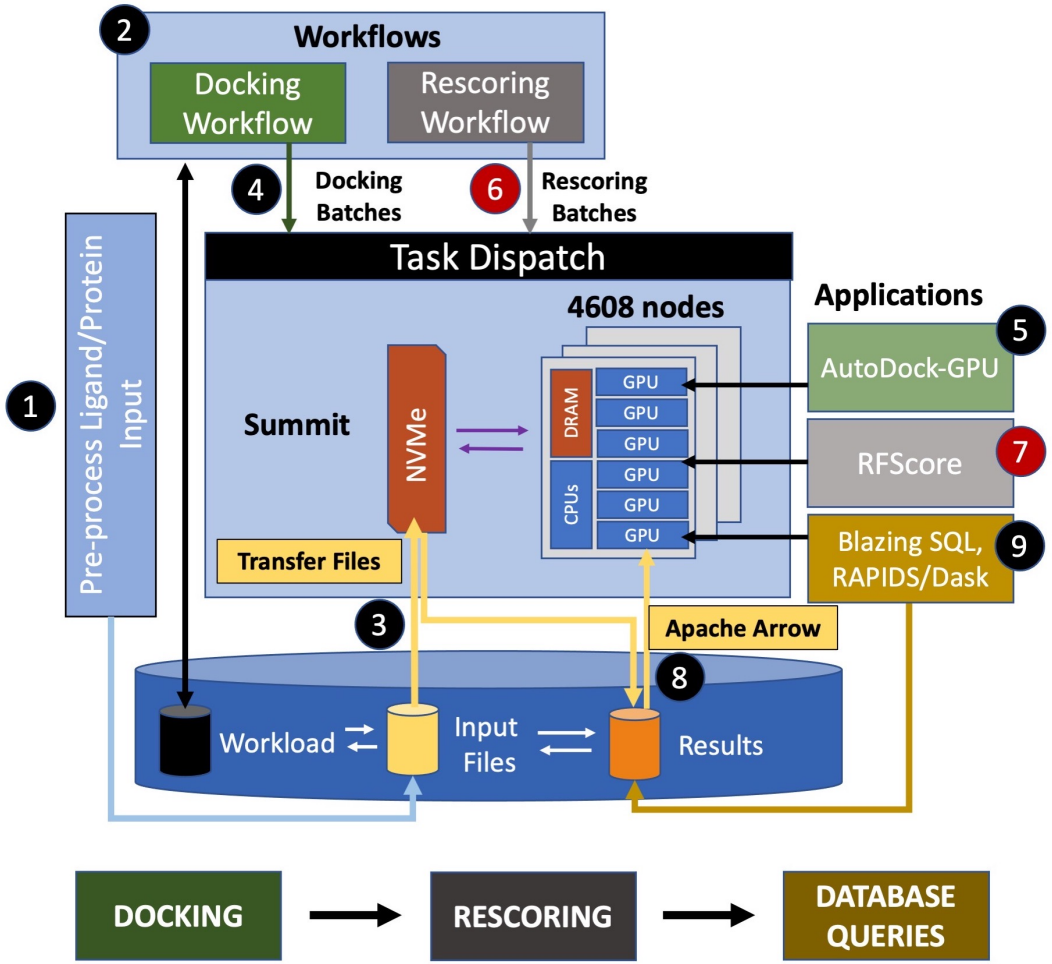
- Mongo-DB hosted on OLCF Slate/Marble Kubernetes
- Fireworker script interacts with task graph
- Largest deployment to date (27,600 fireworkers)
- More components increased development cycle time
- Persistent database state captures provenance data and allows checkpoint/restart
- https://github.com/materialsproject/fireworks

Redis Queue

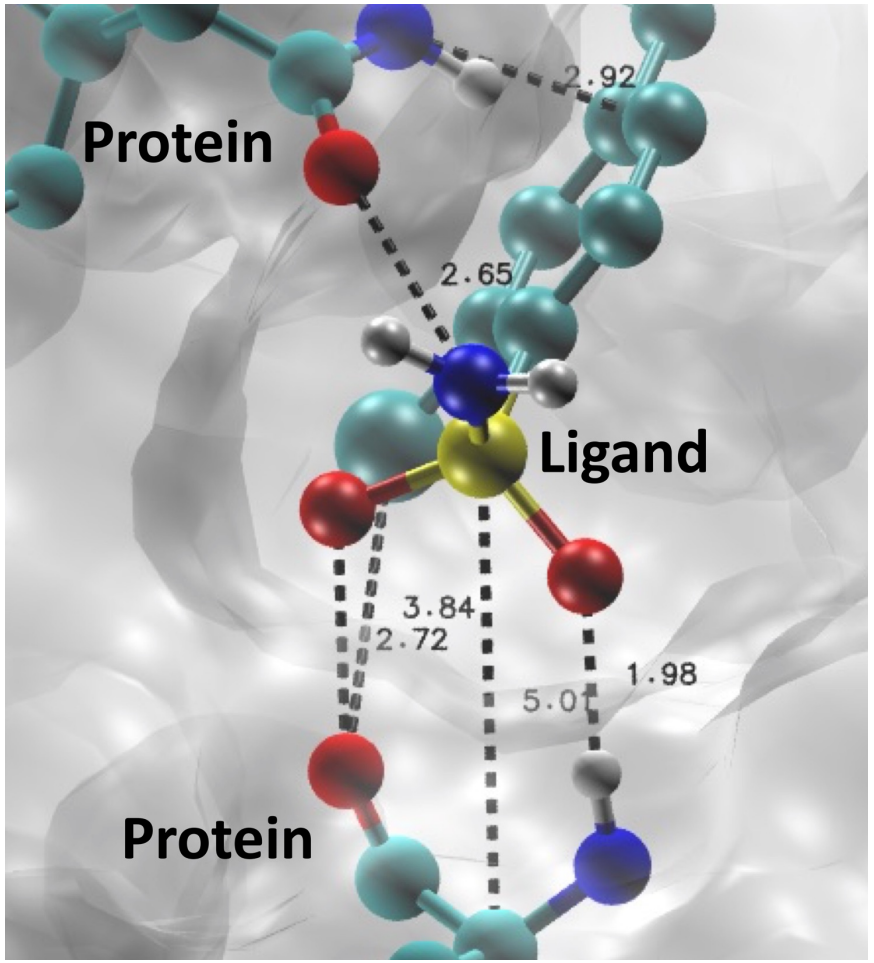
- Redis-DB hosted on job-launch node
- Custom script interacts with ready/complete/error sets
- Small code size, special purpose solution
- Persistent database state allows checkpoint/restart
- Provenance data captured in per-node log
- Simplified resqueue design (https://github.com/resque/resque)

Accelerating the End-to-end Pipeline >50×

1. High-throughput AutoDock-GPU on Summit: Docking Billions of Compounds at Scale for COVID-19 Drug Discovery
2. **Accelerated Kernels for Machine Learning Feature Calculations: Better Predictions via ML-based Rescoring**
3. Data Analytics on Massive Outputs Within a GPU-accelerated Virtual Laboratory



Feature Calculation for Machine Learning



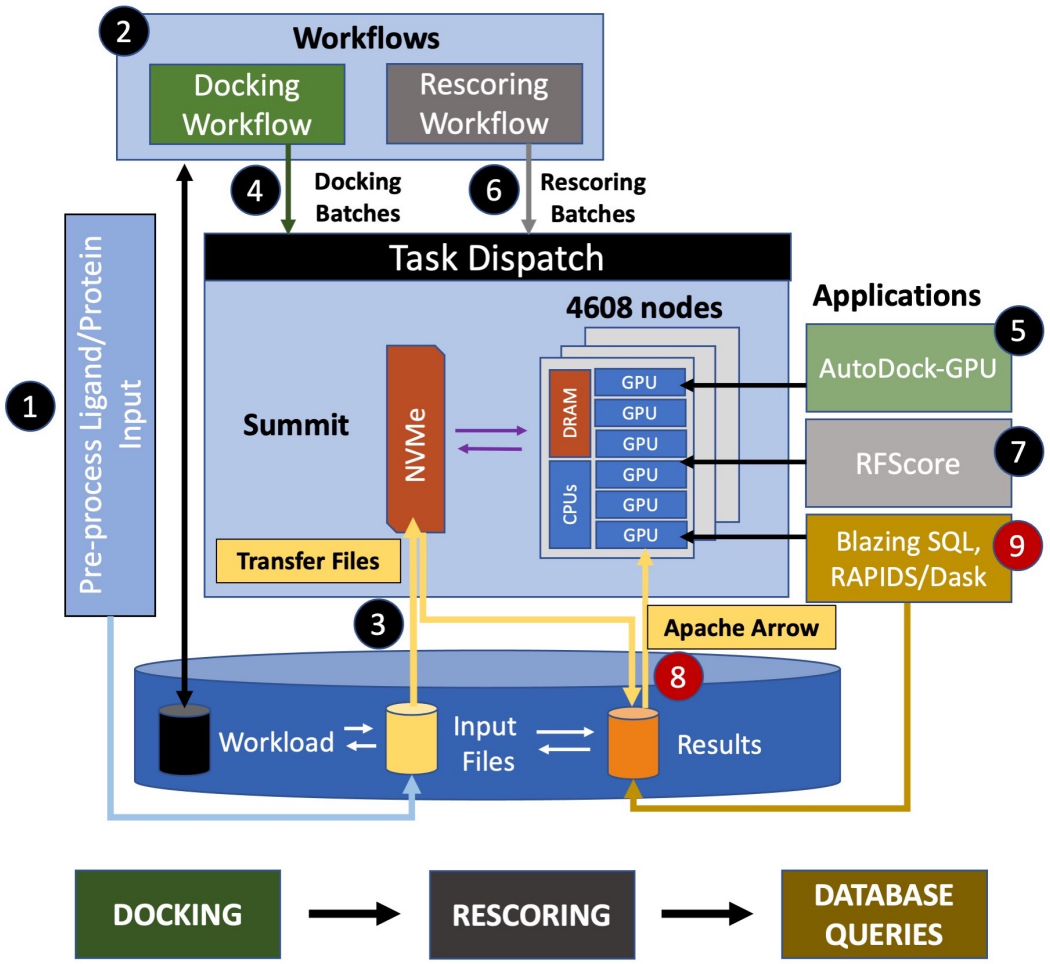
- Implemented a pairwise contact histogram kernel
- Accelerate parsing of PDBQT coordinate files using tokenization on GPU with string methods in `cudf`
- Stream all 20 conformations per ligand from a CUDA data frame to the GPU kernel using zero-copy

	T_{ligand} [ms]	ligs/Ns	$T_{\text{parse},N=72}$	TTS
CPU	271.4	154.76	3.216h	32.407h
GPU	0.387	15,494	0.314h	0.323h
Speedup	**700**×		**10.24**×	**100.12**×

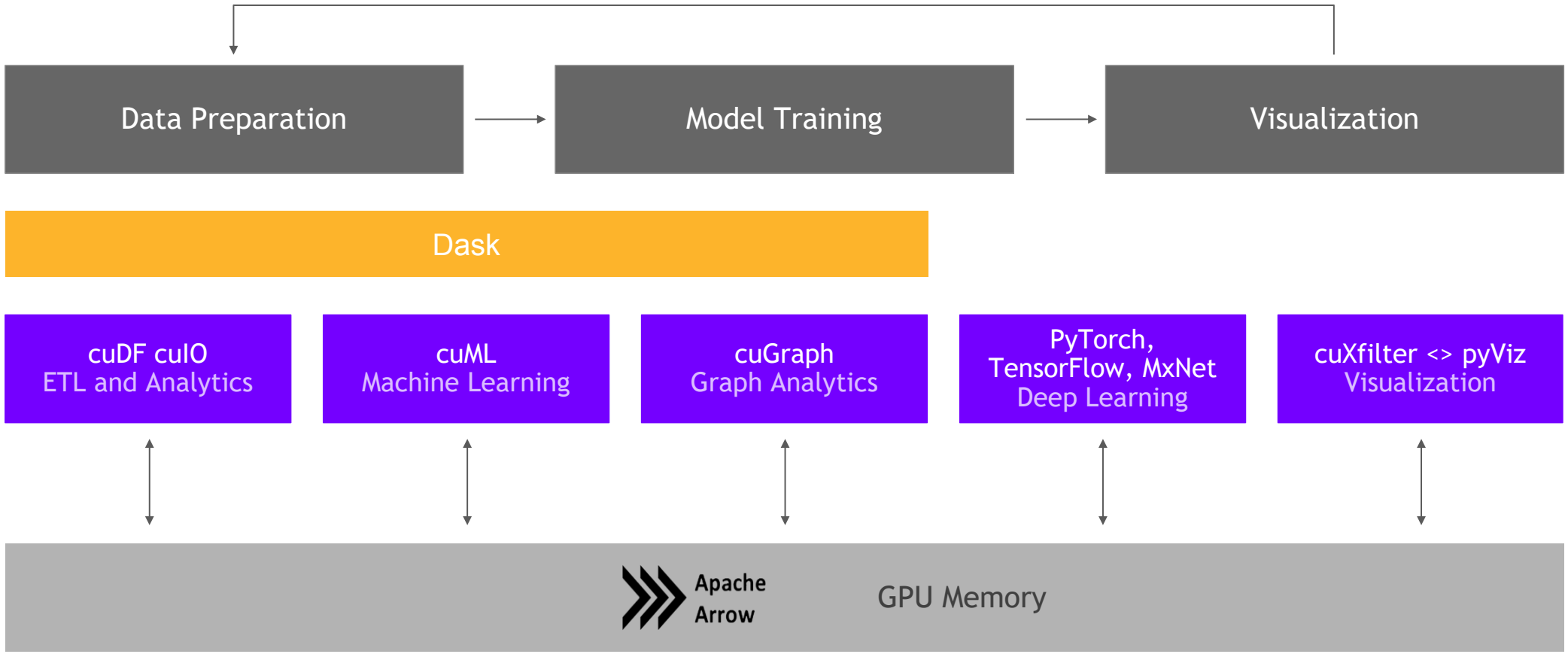
Per-node time to solution,
Rescoring on 72 nodes

Accelerating the End-to-end Pipeline >50×

1. High-throughput AutoDock-GPU on Summit: Docking Billions of Compounds at Scale for COVID-19 Drug Discovery
2. Accelerated Kernels for Machine Learning Feature Calculations: Better Predictions via ML-based Rescoring
3. **Data Analytics on Massive Outputs Within a GPU-accelerated Virtual Laboratory**



Accelerated Data Analytics Key Concepts



Apache Arrow memory management avoids the serialization-deserialization bottleneck

NVIDIA RAPIDS

Accelerated Data Analytics "Virtual Lab" at OLCF

jupyter → DASK →

blazingSQL

Accelerated SQL queries

RAPIDS

1.3 TB of parquet files per docking

- User interface
- IPython-based
- Interactive

- Connection
- Task Scheduling across nodes/GPUs

- Accelerated Data Analytics
- Machine Learning
- Visualization

Front-end cluster with GPUs **(Marble/Slate)**

Back-end Supercomputer with high-memory GPUs + node-local storage **(Summit)**

We are analyzing TBs of data in seconds/minutes using Summit's GPUs

Interactive HPC for Scientific Productivity

Goal: analyze 1.3B docking results in seconds

JupyterLab

jupyterhub-bif128.apps.marble.ccs.ornl.gov/user/2731a8ec-de5f-4de9-9bcd-9a6baed053f9/lab/workspaces/auto-A

File Edit View Run Kernel Tabs Settings Help

Dask Gpu Memory

GPU Memory: 2.24 TB / 16.44 TB

0.0 2.0 GiB 4.0 GiB 6.0 GiB 8.0 GiB 10.0 GiB 12.0 GiB 14.0 GiB

Dask Gpu Utilization

GPU Utilization

0 20 40 60 80 10

Dask Nprocessing

Tasks Processing

0 0.2 0.4 0.6 0.8 1

Dask Task Stream

Task Stream

1/01 10s 20s 30s 40s

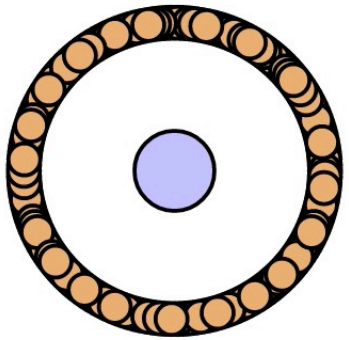
5R84_analyze.ipynb

Code Python 3

Visualization

```
[ ]: !obabel -ipdbqt 'test_5R84.pdbqt' -opdb -O 'test.pdb'

[34]: mport py3Dmol
      igand = open('test.pdb','r').read()
      eceptor = open('monomer_5R84.pdb','r').read()
      iew = py3Dmol.view(width=480,height=600)
      iew.addModel(receptor)
      iew.setStyle({'chain': 'P'},{'cartoon': {'color':'spectru
      view.setStyle({'chain': 'P'},{'stick': {'color':'spectrum
      iew.addSurface(py3Dmol.VDW,{'opacity':0.4,'color':'white'
      iew.addModel(ligand)
      iew.setStyle({'model': 1},{'stick':{'radius':0.5,'singleB
```

Dask Cluster Map



6WQF_analyze.ipynb

Code Python 3

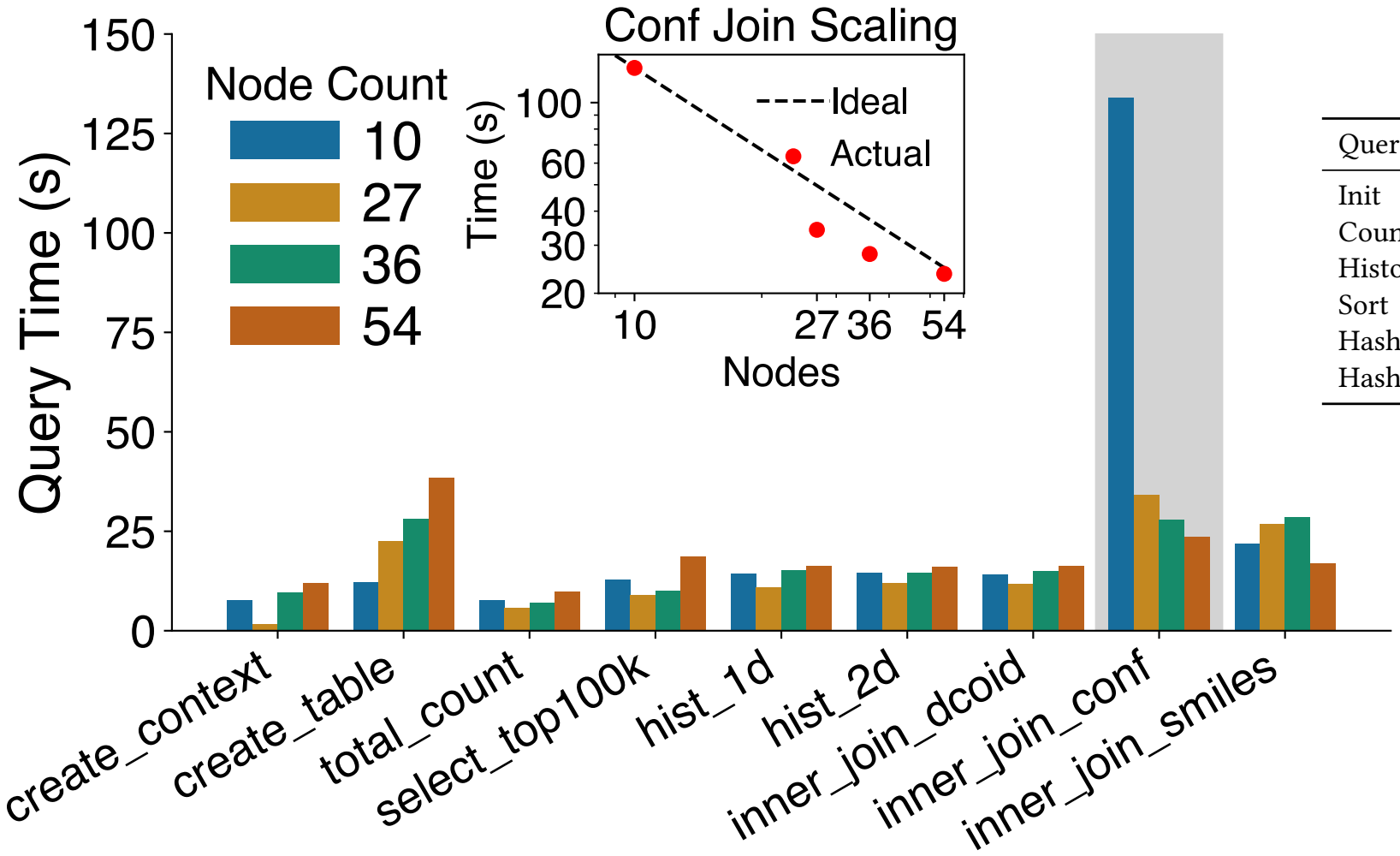
Select top compounds

```
[51]: q = "SELECT * FROM scores ORDER BY vs_dude_v2 DESC LIMIT 100000"

[52]: bc.explain(q)

[52]: 'LogicalSort(sort0=[$4], dir0=[DESC], fetch=[100000])\n  LogicalTableScan
      (table=[[main, scores]])\n'

[53]: %%time
      res = bc.sql(q)

      CPU times: user 571 ms, sys: 27.1 ms, total: 598 ms
      Wall time: 11.2 s
```

1 7 Python 3 | Idle Saving completed Mode: Edit Ln 3, Col 8 5R84_analyze.ipynb

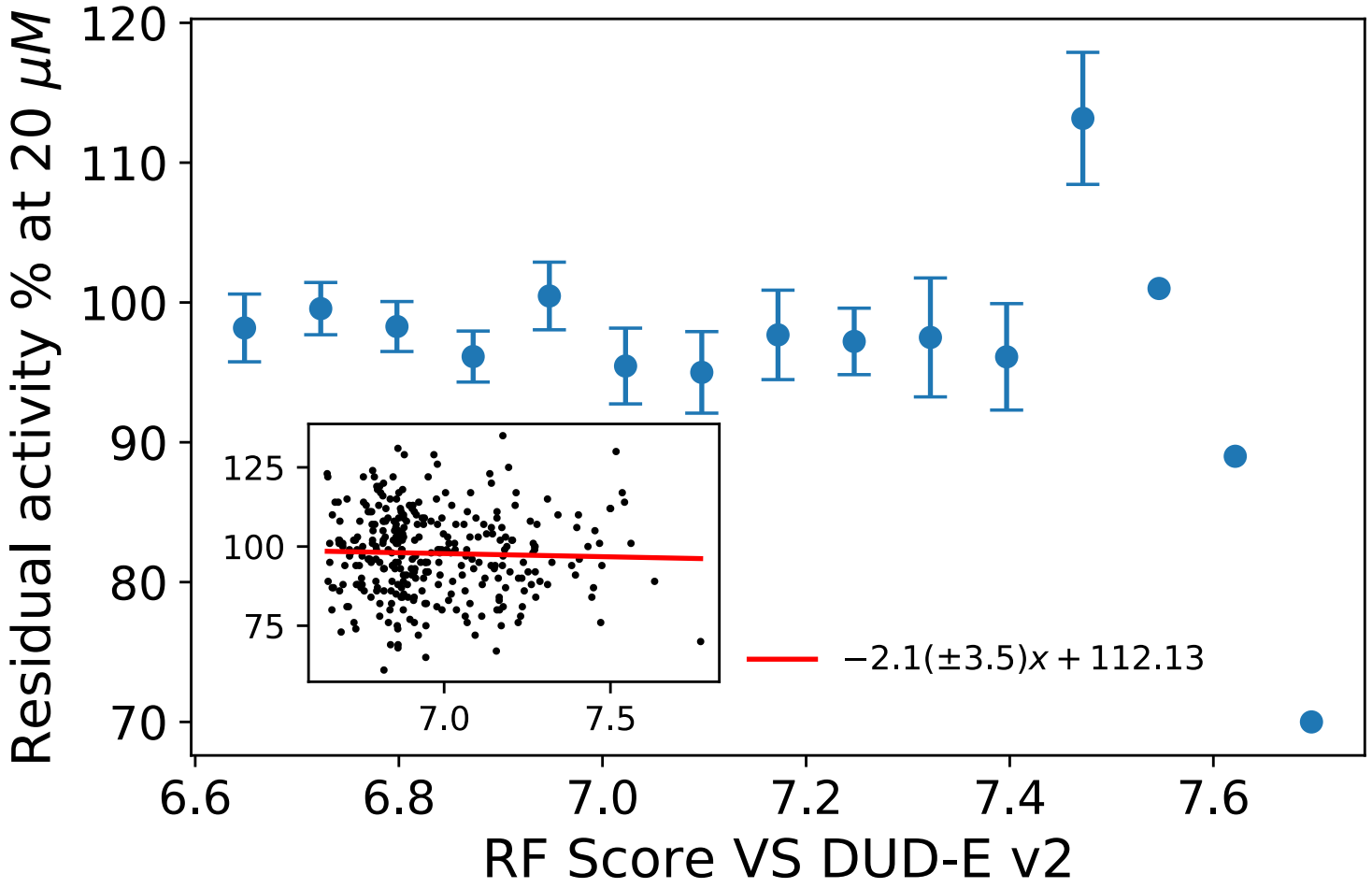
Demo: Virtual Lab using Summit's GPUs

Strong scaling of database query processing



Query Type	complexity
Init	$\mathcal{O}(1)$
Count	$\mathcal{O}(N)$
Histogram	$\mathcal{O}(N)$
Sort	$\mathcal{O}(N \log N)$
Hash join (partition)	$\mathcal{O}(N + M)$
Hash join (join)	$\mathcal{O}\left[(N/N_{\text{part},N}) \times (M/N_{\text{part},M})\right]$

From Virtual Lab to Wet-lab Experiments

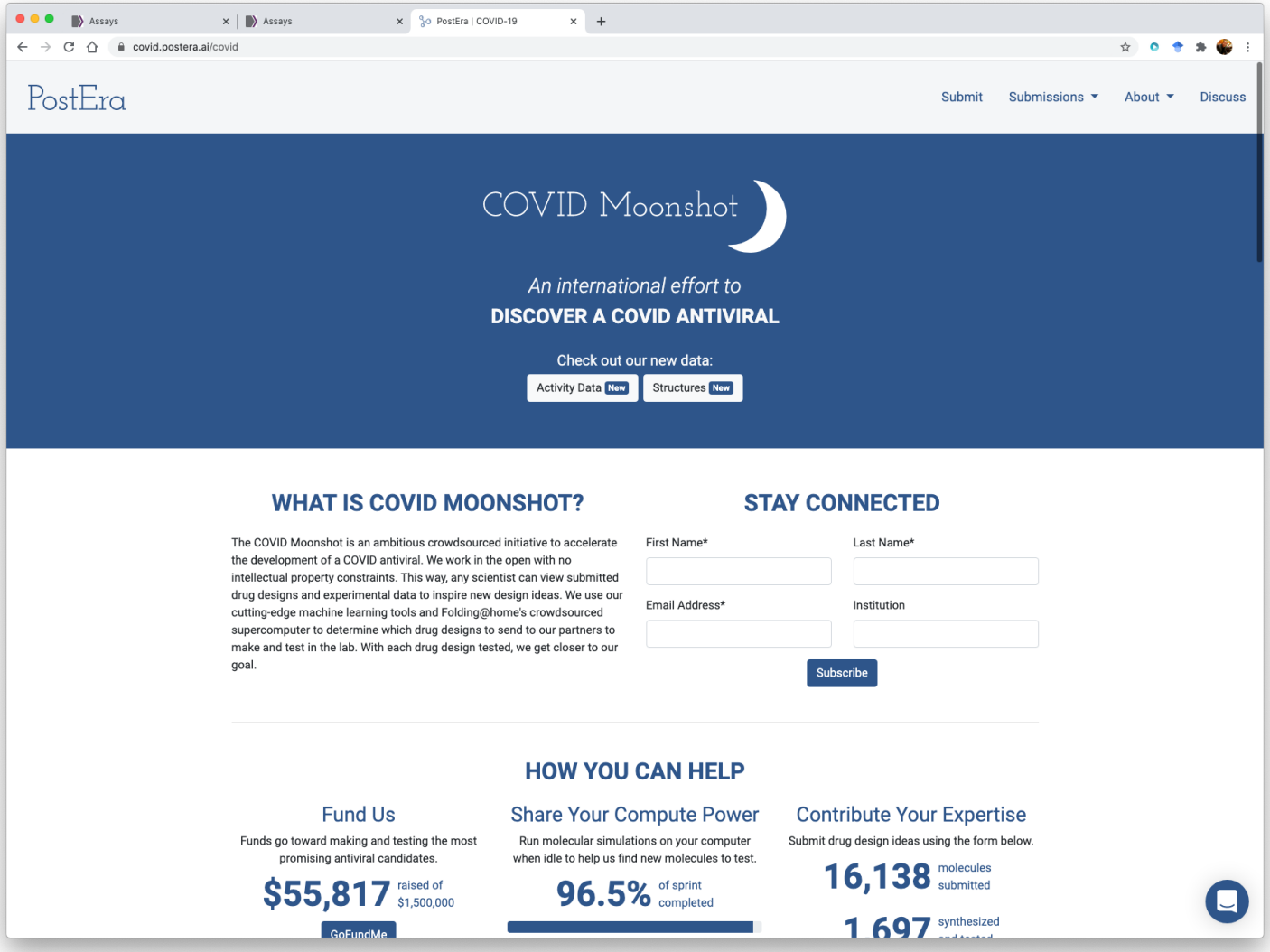


Purchased and assayed 323 compounds predicted by Gigadocking

(testing by Stephanie Galanie, ORNL)

Goal: Validate top compounds from 1.3B Gigadocking predicted via the "Virtual Lab"

Postera.ai "Moonshot" Mpro data set



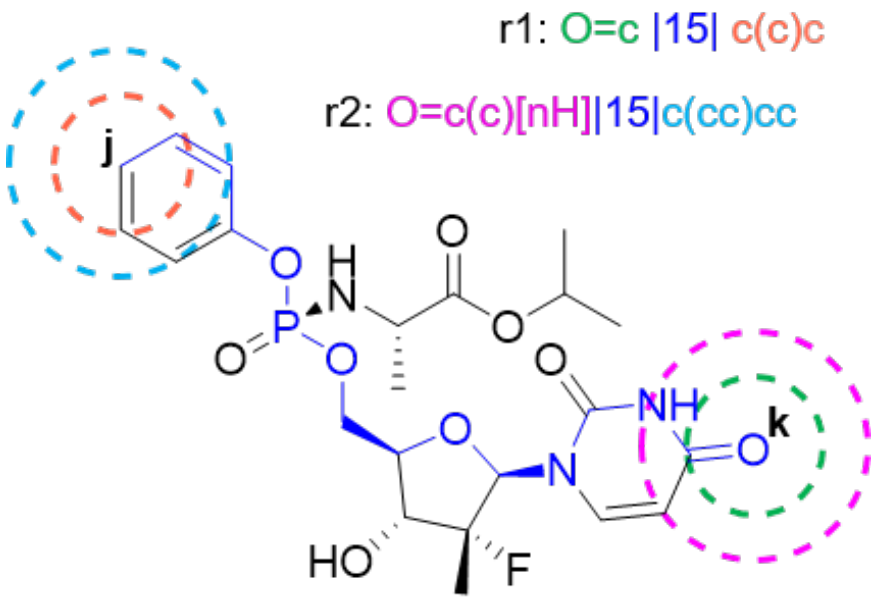
"*Crowd-sourcing*" effort

~1000 activities
~500 crystal structures

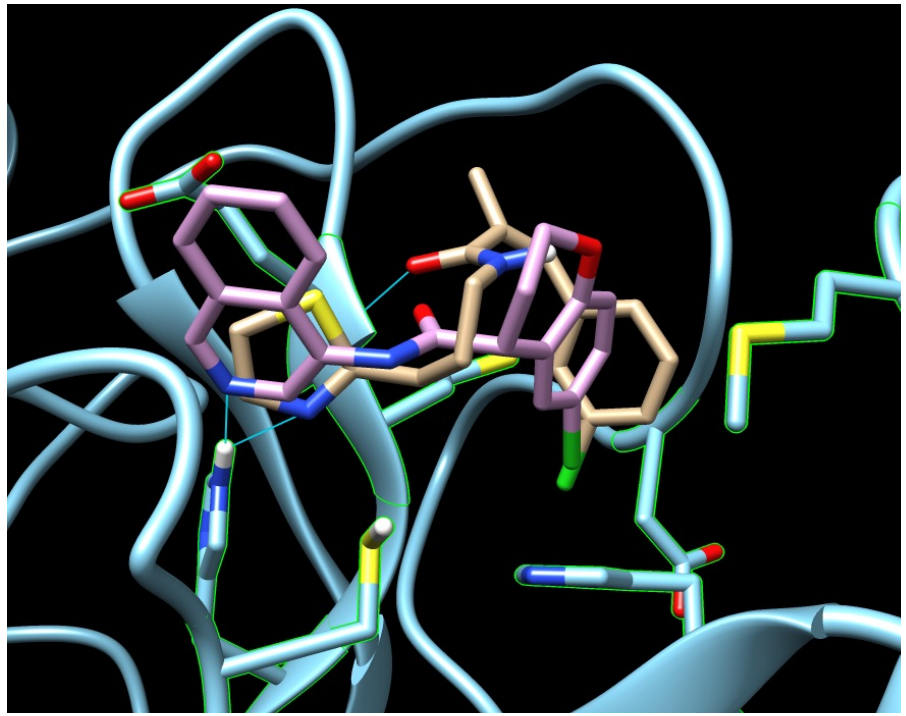
https://covid.postera.ai/covid

Hit Expansion for SARS CoV-2 Main Protease

- We *computationally* predicted **new non-covalent inhibitors** that
  1. maximize **scaffold similarity** to a known inhibitor
  2. make similar **docking contacts with the protein target**



2D neighborhood expansion (MAP4)



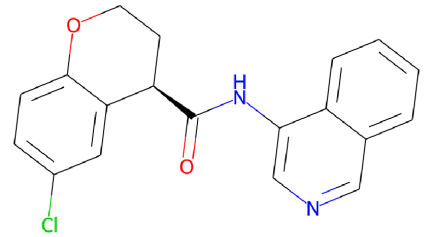
3D contact expansion (giga-docking)

Moonshot x11612 (xtal) and Z1528050012 (docked)

Capecchi, A., Probst, D. and Reymond, J.L., 2020. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. *Journal of Cheminformatics*, *12*(1), pp.1-15

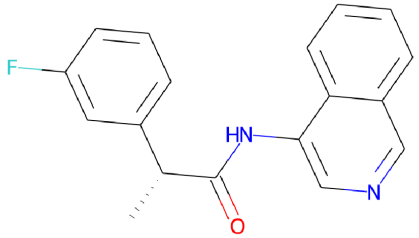
Hit Expansion for SARS CoV-2 Main Protease

- Purchased 100 Molecules from Enamine
- Experimental screen for activity against SARS CoV-2's main protease

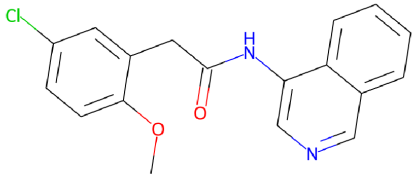


x11612 0.0% (0.000)

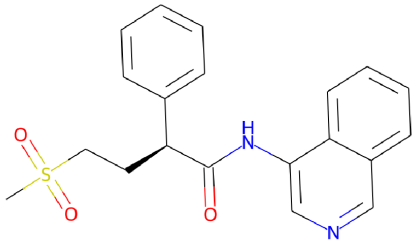
Reference compound from COVID Moonshot
(not assayed)



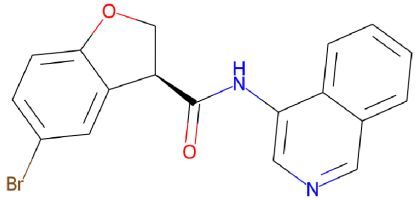
Z1530724963_2_T1 9.0% (0.285)



Z1530718726_1_T1 9.0% (0.302)



Z1927517858_1_T1 10.0% (0.287)



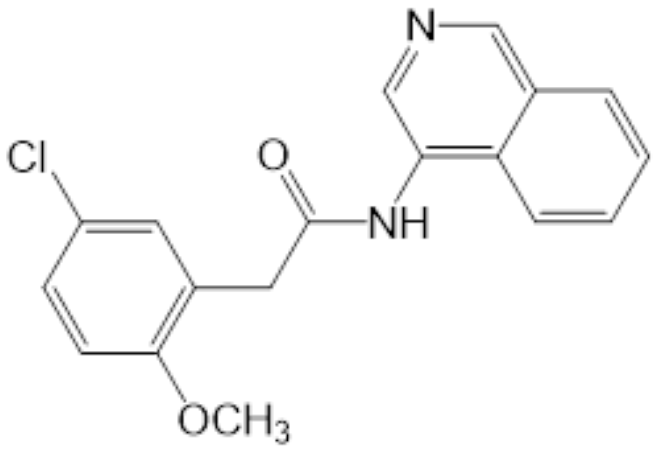
Z1530725178_1 10.0% (0.365)

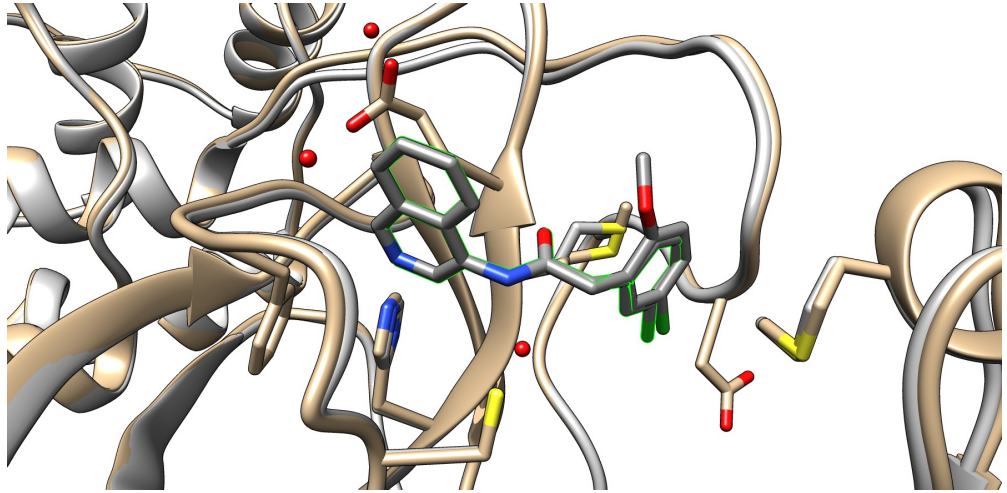
Better Inhibition

Unpublished data

New Crystal Structures from Hit Expansion

- Three compounds were purchased from Enamine and protein structures solved at ORNL's SNS facility
- Two of the compounds co-crystallized with the main protease and characterized using X-ray crystallography at room temperature
- Three additional compounds are undergoing crystallization trials

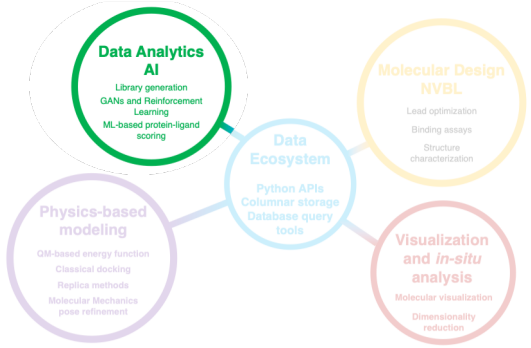




Z1530718726

Unpublished data

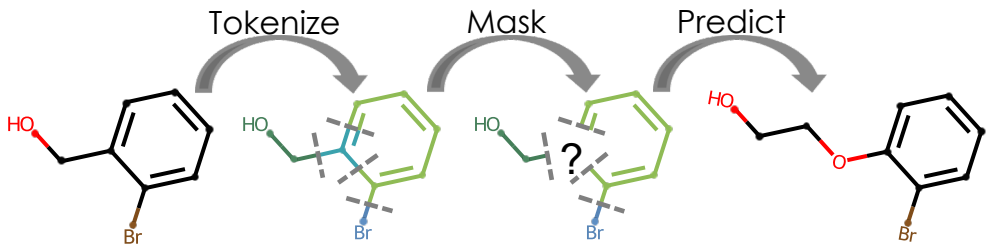
Crystallization/Characterization: Daniel Kneller and Andrii Kovalevskyi (ORNL)



New Language Model for Molecules

We developed a state-of-the-art, transformer-based machine learning model to predict novel, synthesizable compounds

Manipulating SMILES to create new molecules

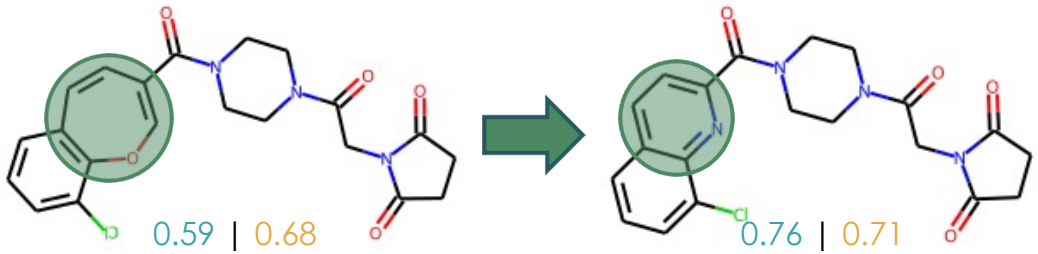


OC/c1ccccc1Br

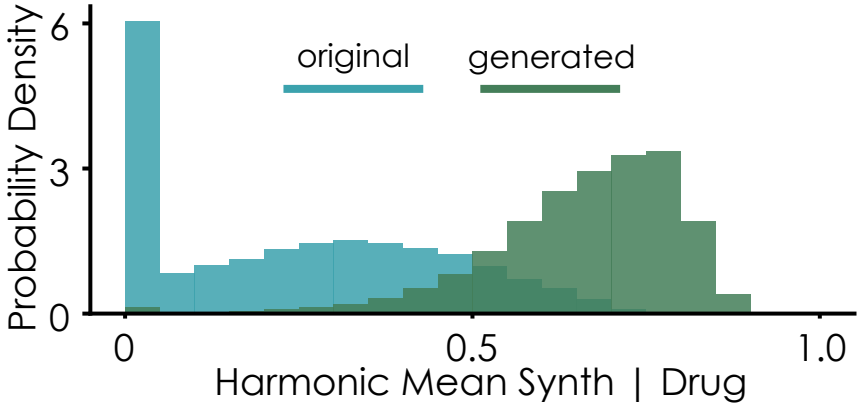
OC/COc1ccccc1Br

Example for a single molecule

Optimizing synthesizability and drug-likeness



Results across a population



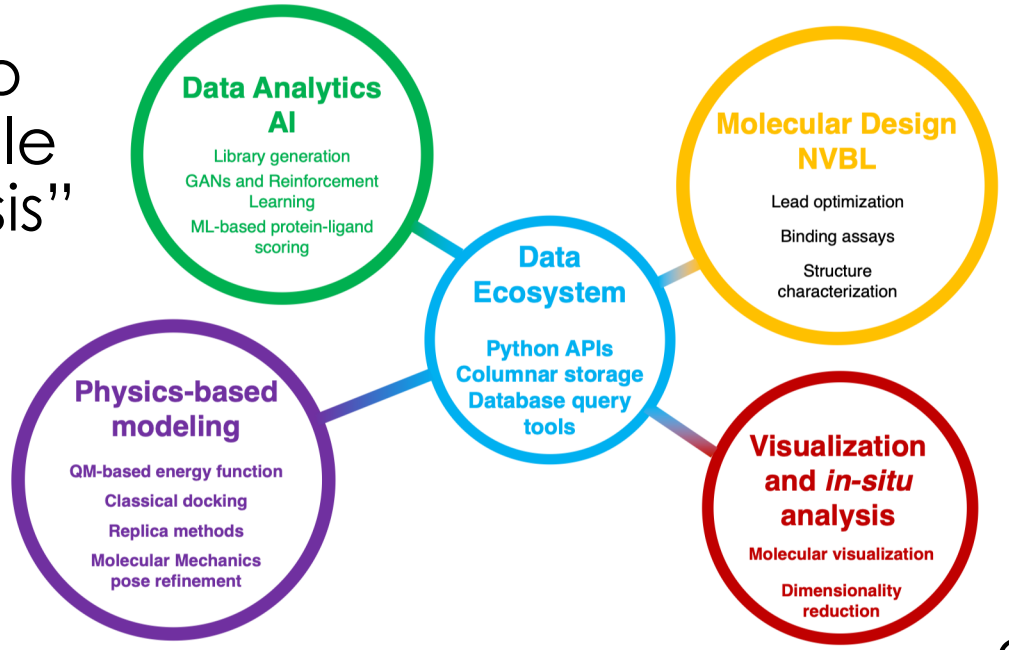
https://code-int.ornl.gov/ueq/maskedevolution

Conclusion

- A computational capability to accelerate the initial stages of drug discovery is essential to **combat the current and future pandemics**
- We performed a **virtual screening of 1.37B small organic molecule compounds on all of Summit** against the SARS CoV-2 main protease in under 24h and **predicted novel inhibitors of Mpro**

In-silico molecule "synthesis"

Ongoing



Biophysical characterization

Data analytics

High-throughput ↔ High-Accuracy Virtual Screen