NAVIGATING THE MOLECULAR MAZE: A PYTHON-POWERED APPROACH TO VIRTUAL DRUG SCREENING

Johnny Raicu

Advisors: Valerie Hayot, Kyle Chard, Ian Foster

University of Chicago

November 15, 2023

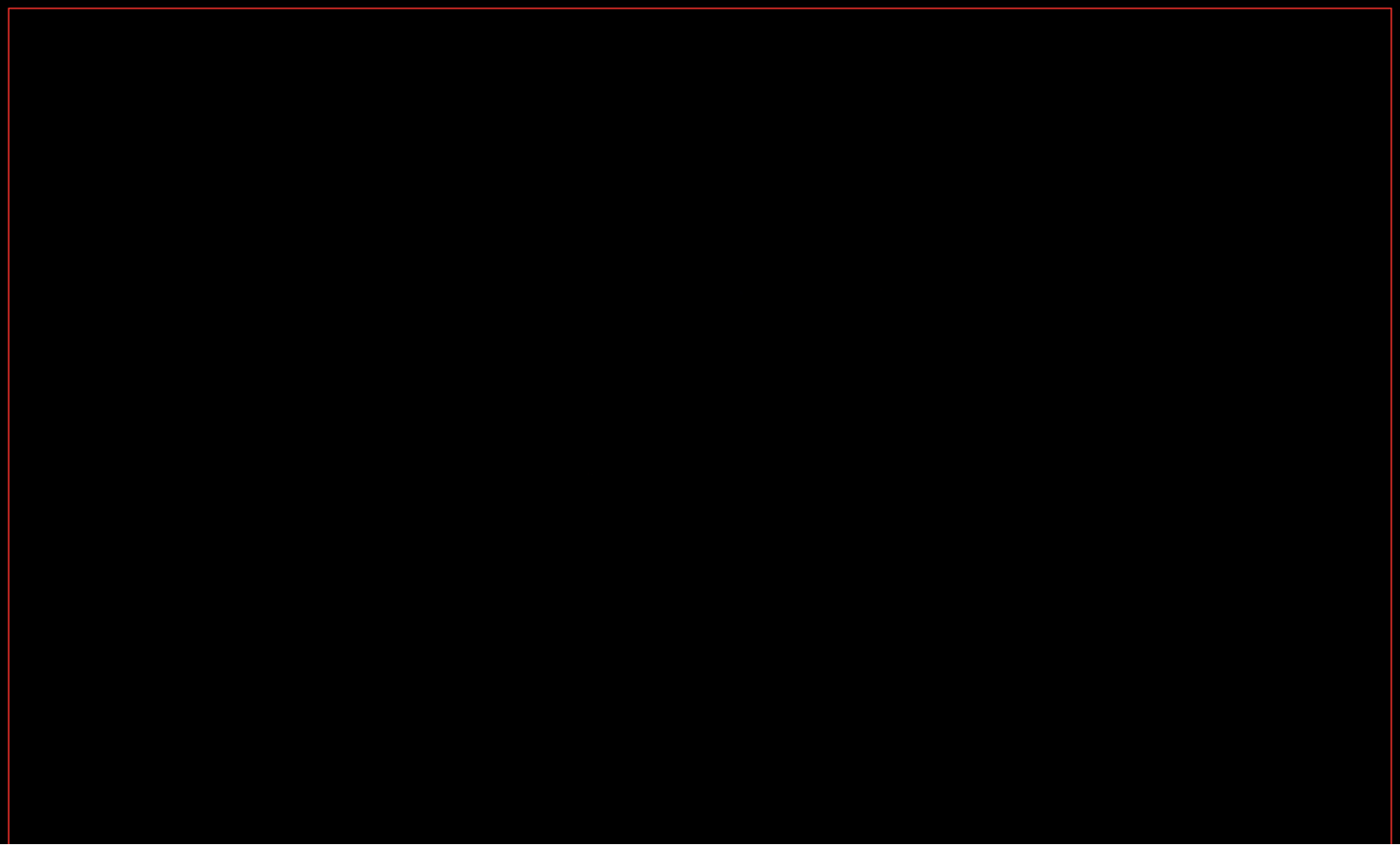
ACKNOWLEDGEMENTS

- Research work performed in summer of 2023
  - Globus Labs at University of Chicago
- Mentors
  - Valerie Hayot-Sasson
  - Kyle Chard
  - Ian Foster

OVERVIEW

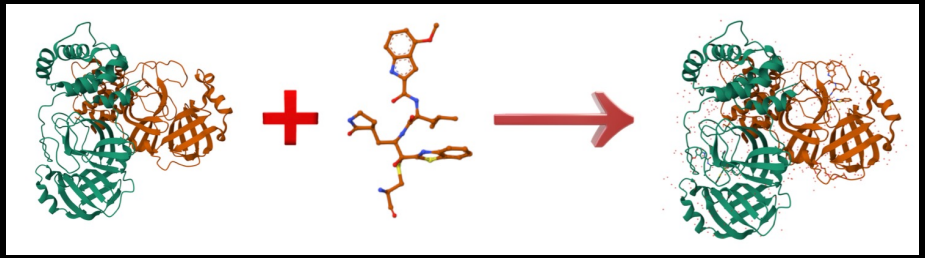
- COVID-19 pandemic brings urgency to accelerating virtual drug screening
- Brute force molecular search space is enormous
  - Finding the optimal drug candidates involves millions of Monte Carlo simulations that typically run for hundreds of seconds each ➔ thousands of years of computing on a single core
- Lightweight machine learning surrogate models can reduce the search space between 10x and 100x while maintaining high accuracy

WHAT IS PROTEIN DOCKING?



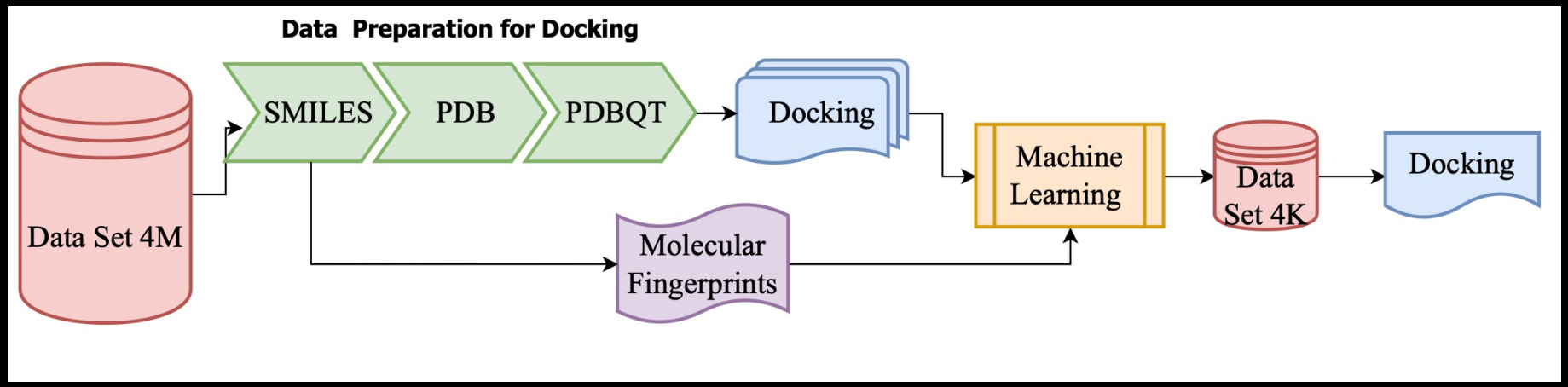
PROBLEM STATEMENT

- What is the problem?
  - Identifying the "best" ligands from a dataset of molecules by combining simulation and ML algorithms on HPC resources



- What are the challenges?
  - Machine learning model accuracy
  - Sampling efficiency
  - Computational cost
  - Complexity of docking workflow

PROPOSED SOLUTION

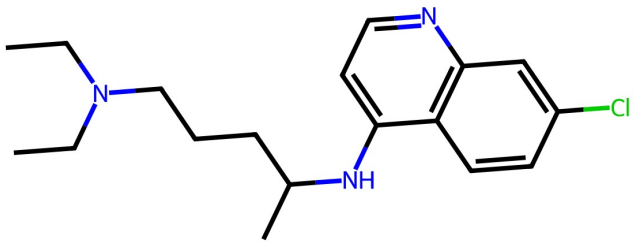


SMILES AND FINGERPRINTS

Hydroxychloroquine SMILES String

Example:
CCN(CCCC(C)NC1=C2C=CC(=CC2=NC=C1)Cl)CCO

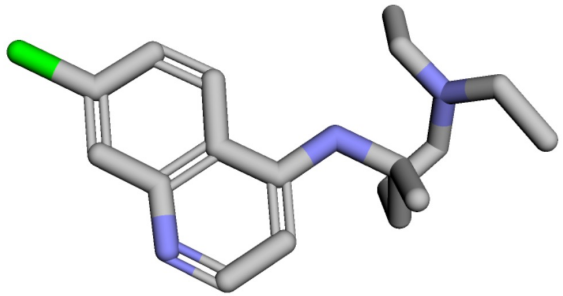
Explanation:
The simplified molecular-input line-entry system (SMILES) uses chemical notation to represent the structure of a molecule visualized in 2D below.



Hydroxychloroquine Fingerprint

Example:
1110010011110101111100111101101101111111001111111100001001101 01

Explanation:
Molecular fingerprints are bit-vectors that help a machine learning model map a molecule description to a docking score.



TOOLS, LIBRARIES, AND DATASETS

- **Programming Tools**
  - Python 3.8.3, Parsl 1.3.0.dev0
- **Libraries**
  - AutoDock Vina 1.2.3, Visual Molecular Dynamics 1.9.3, Scikit-learn 1.3.0, NumPy 1.24.3, Pandas 1.5.3
- **Dataset**
  - 0.9 GB file containing four million ligands stored as SMILES strings

SMILES ➔ PDB ➔ PDBQT

- **DB03048 SMILES**: c1c([nH]c(=O)[nH]c1=O)CC(=O)[O-]
- **DB03048 PDBQT**:

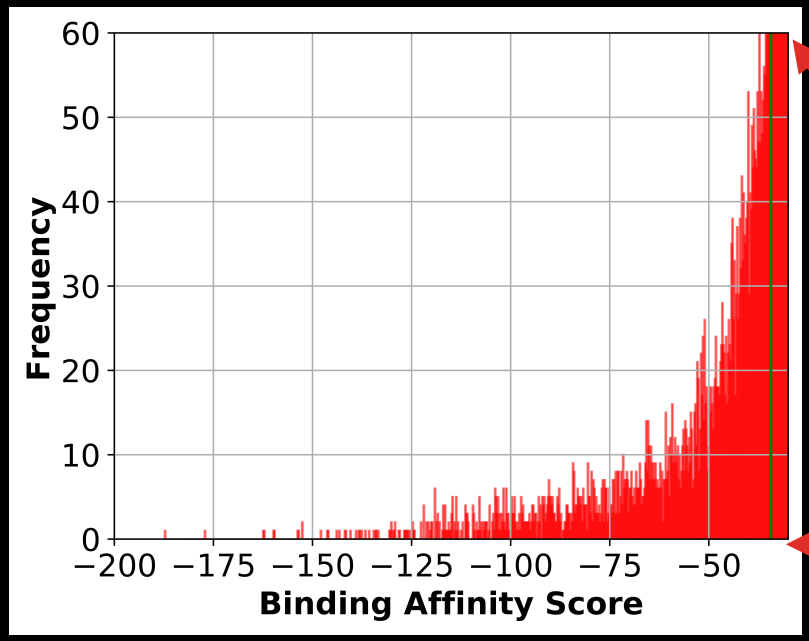
```
REMARK  2 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK    1  A    between atoms: C2_2  and  C5_9
REMARK    2  A    between atoms: C5_9  and  C6_10
ROOT
ATOM      1  C1  UNL X   1      -0.335   1.388  -0.190  1.00  0.00     0.100 A
ATOM      2  C2  UNL X   1       0.260   0.195  -0.322  1.00  0.00     0.029 A
ATOM      3  N1  UNL X   1      -0.405  -0.944   0.070  1.00  0.00    -0.312 N
ATOM      4  C3  UNL X   1      -1.656  -0.960   0.616  1.00  0.00     0.327 A
ATOM      5  O1  UNL X   1      -2.225  -1.982   0.991  1.00  0.00    -0.247 OA
ATOM      6  N2  UNL X   1      -2.264   0.262   0.718  1.00  0.00    -0.275 N
ATOM      7  C4  UNL X   1      -1.696   1.454   0.373  1.00  0.00     0.251 A
ATOM      8  O2  UNL X   1      -2.276   2.528   0.510  1.00  0.00    -0.268 OA
ATOM      9  H2  UNL X   1       0.132  -1.821   0.056  1.00  0.00     0.170 HD
ATOM     10  H3  UNL X   1      -3.183   0.278   1.125  1.00  0.00     0.173 HD
ENDROOT
BRANCH   2  11
ATOM     11  C5  UNL X   1       1.637   0.025  -0.910  1.00  0.00     0.170 C
BRANCH  11  12
ATOM     12  C6  UNL X   1       2.526  -0.888  -0.063  1.00  0.00     0.178 C
ATOM     13  O3  UNL X   1       2.096  -2.074   0.067  1.00  0.00    -0.648 OA
ATOM     14  O4  UNL X   1       3.580  -0.357   0.385  1.00  0.00    -0.648 OA
ENDBRANCH  11  12
ENDBRANCH   2  11
TORSDOF 2
```

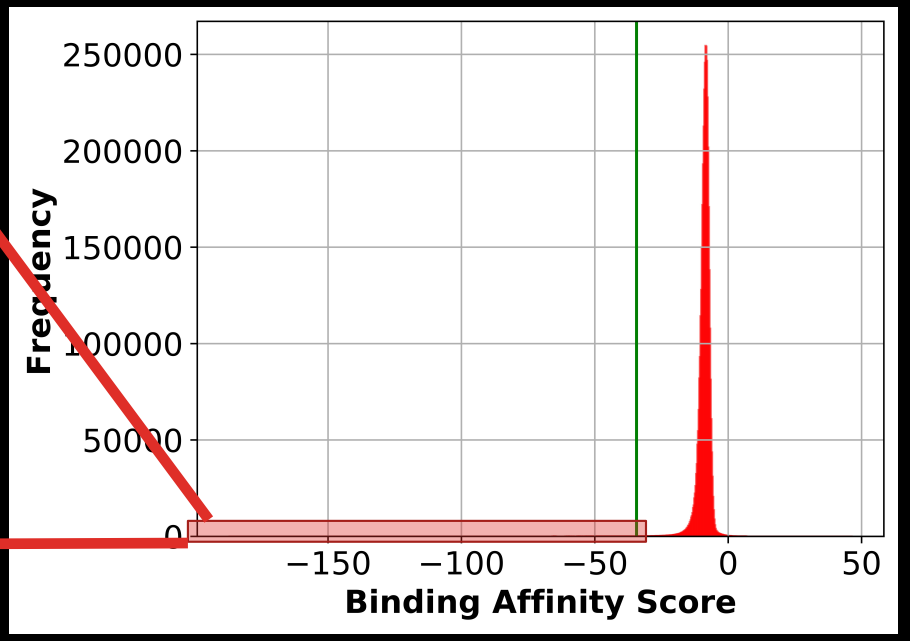
DOCKING

Ligand (DB03048) to Receptor (1iep)

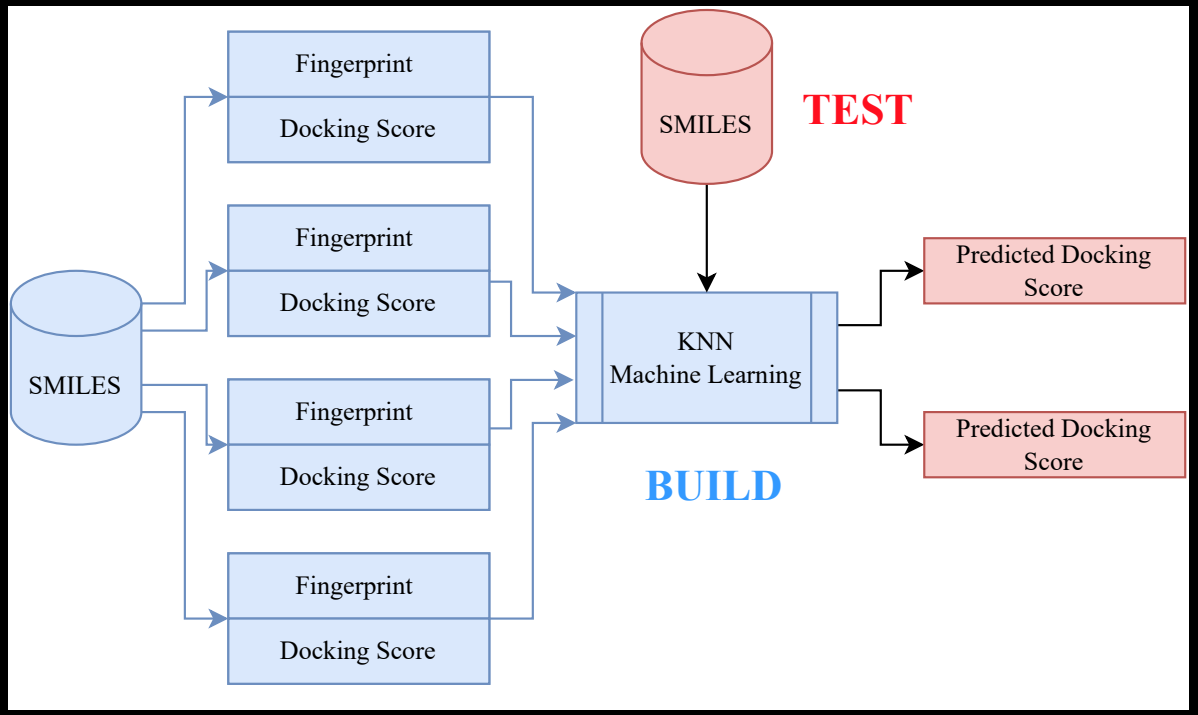
```
Scoring function : vina
Rigid receptor: 1iep_receptor.pdbqt
Ligand: DB03048-0.pdbqt
Grid center: X 15.614 Y 53.38 Z 15.455
Grid size  : X 20 Y 20 Z 20
Grid space : 0.375
Exhaustiveness: 32
CPU: 32
Verbosity: 1

Computing Vina grid ... done.
Performing docking (random seed: 1849697511) ...
0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
**************************************************

mode |   affinity | dist from best mode
     | (kcal/mol) | rmsd l.b.| rmsd u.b.
-----+------------+----------+----------
   1        -6.36          0          0
   2       -6.298      2.329      4.576
   3       -6.174        2.1      2.931
   4       -6.069     0.9309      1.116
   5       -5.979       1.92      4.843
   6       -5.974      1.879      4.509
   7        -5.96      1.979      4.329
   8       -5.932      2.677       4.55
   9       -5.896      2.014      3.215
  10       -5.827      1.756      2.051

3.1626994609832764: docked 1/1 1iep_receptor.pdbqt to DB03048 -6.36
dock DB03048 3.1565709114074707 -6.36
Elapsed time run:  3.1630148887634277 seconds
```

DOCKING SCORE DISTRIBUTION



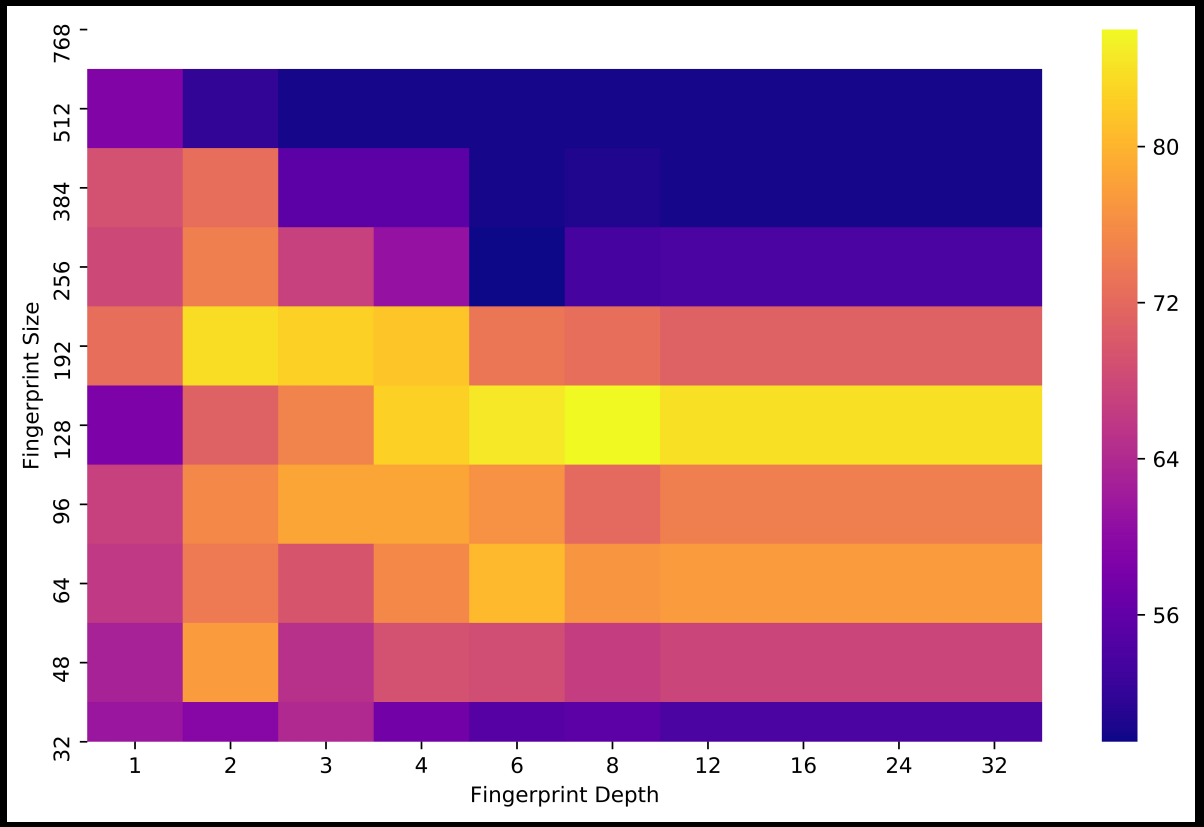


MACHINE LEARNING



OPTIMIZING MACHINE LEARNING PARAMETERS

- Depth
  - Radius Parameter
  - Local chemical environment
  - Larger radius captures interactions
- Size
  - Number of bits
  - Level of granularity of molecular structure

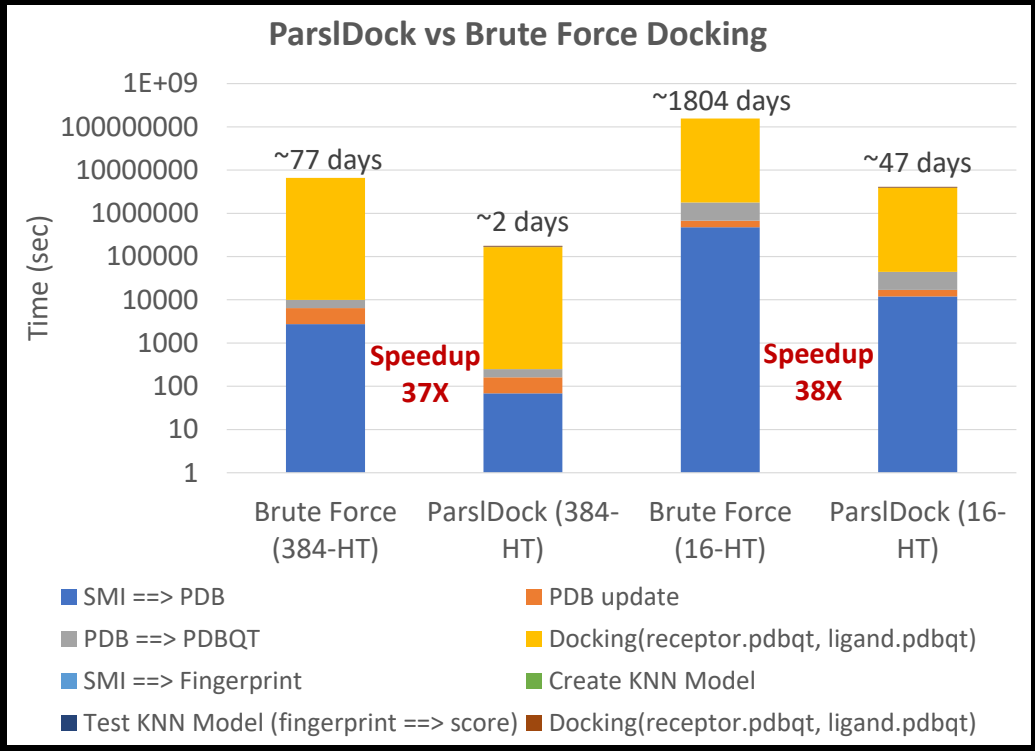


TESTBEDS

- 8c-laptop:
  - 8-core Intel Core i9 CPU 2.4GHz
  - 64GB DDR4
  - 8TB NVMe
  - MacOS 12.6.3
- 192c-server
  - 8x 24-core x86 Intel Xeon CPU 2.1GHz
  - 786GB DDR4
  - 16TB SSD
  - Ubuntu Linux 22.04

PARSLDOCK PERFORMANCE

- Up to 38X speedup on ParslDock vs. Brute Force Docking
- Linear scalability from 8-core laptop to 192-core server
- Docking (yellow) consumes the most compute time at 97%



ONGOING AND FUTURE WORK

- ParslDock to showcase Parsl support for fine grained parallelism
  - Joint work with Jamison Kerney (IIT) aiming for future conference paper submission
- KNN relies on accurate distance metrics between samples
  - Explore various types of distance measures: Jaccard Coefficient, Tanimoto, Hamming Distance
- Explore additional ML models: deep neural networks

CONCLUSION

- ParslDock: A Python-powered automated pipeline that uses Parsl and machine learning to accelerate the docking process, efficiently utilize compute resources, and reduce the time to discovery
- ParslDock achieves 38X speedup in performance that **makes it possible to execute the virtual drug screening pipeline on a personal computer**
- ParslDock is the basis of a hands-on tutorial used in the Parsl community


Contact me:
johnny.raicu@gmail.com

NOVELTY

- Many works use machine learning to reduce search spaces of applications other than protein docking
  - Material science, Energy Storage
- For works that combined machine learning with virtual drug design
  - Molecular descriptors versus Morgan fingerprints
  - Deep Neural Networks versus KNN

PARSLDOCK PERFORMANCE DETAILS

Stage	# of Tasks	Time/task (sec)	Parallelism	Total Time (sec)
SMI ==> PDB	100000	0.264	384	69
PDB update	100000	0.352	384	92
PDB ==> PDBQT	100000	0.340	384	88
Docking(receptor, ligand)	100000	634.459	384	165224
SMI ==> Fingerprint	4000000	0.001	1	1425
Create KNN Model	200	0.019	1	4
Test KNN Model (fingerprint ==> score)	4000000	0.368	384	3836
Docking(receptor, ligand)	4000	634.459	384	6609