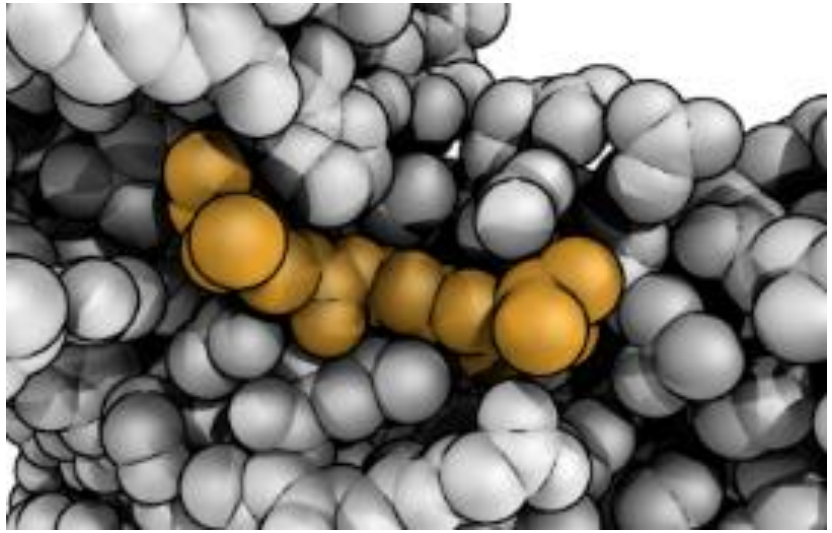
OSG School Showcase
July 15, 2026

Scaling Virtual Screening to Ultra-Large Virtual Chemical Libraries

Spencer Ericksen
Scientist II, Teaching Faculty I

UW Carbone Cancer Center
Drug Development Core
Small Molecule Screening Facility
(interim co-manager)

sericksen@wisc.edu



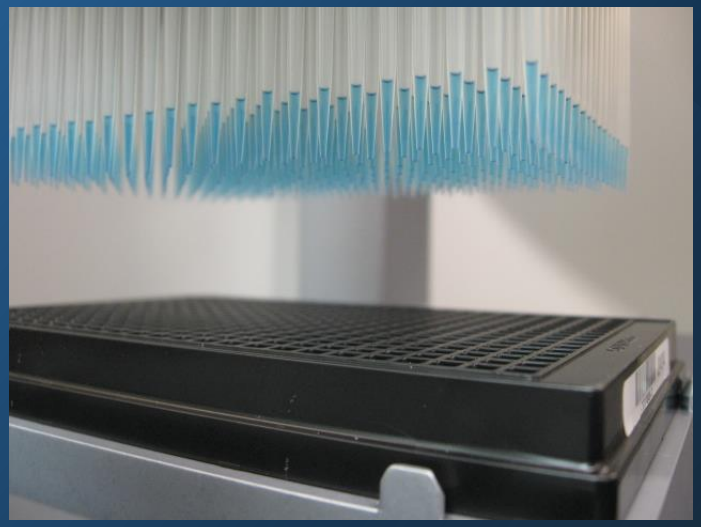
Carbone Cancer Center
UNIVERSITY OF WISCONSIN
SCHOOL OF MEDICINE AND PUBLIC HEALTH

Open Science Grid

CENTER FOR HIGH THROUGHPUT COMPUTING

HTCondor
High Throughput Computing

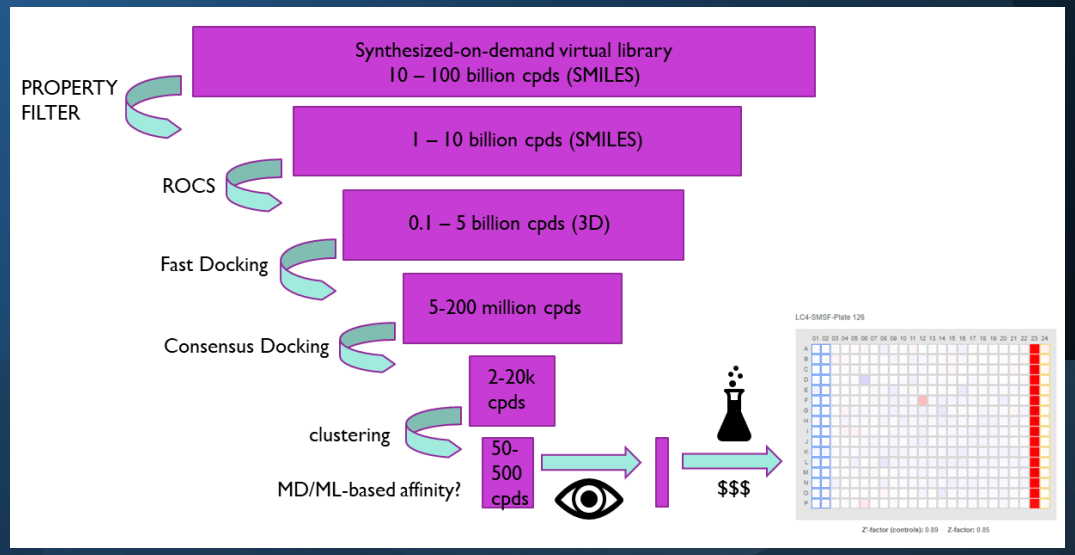
Early-stage drug discovery



- Find rare molecules that affect a specific biological process. Develop as probes or drug candidates.
- Early-stage drug discovery is a needle-in-the-haystack problem—could be 10^{33} drug-like organic molecules.*
- High-Throughput Screening (HTS) is too expensive.

*Polishchuk PG, et al., JCAMD 2013 27(8):675-9

What is Virtual Screening?



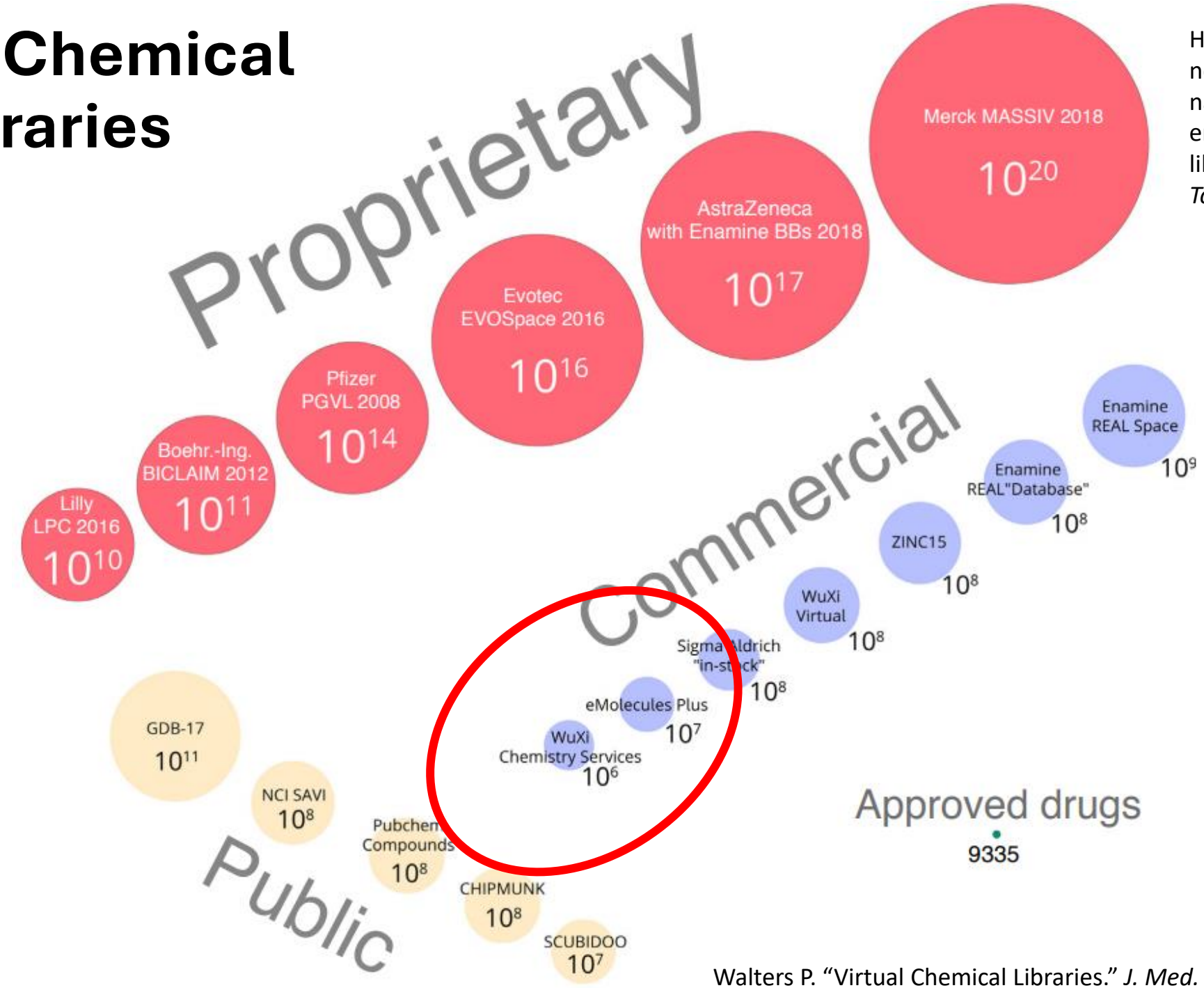
- Virtual Screening: use a computer model to predict “active” molecules within large molecule sets.
- Structure-Based VS uses physics-based (or AI) model to predict whether molecule will bind target protein
- Ligand-Based VS uses supervised ML model to relate molecule structure to a property.
- Goal: reduce number of molecules that must be tested

Real Screening (HTS) vs Virtual Screening(VS)

- test 10^{4-6} cpds
- generates valuable real data
- expensive
- noisy
- can’t scale to ultra-large libraries
- assay must scale to 10^{4-6}

- Virtually screen 10^{8-12} → test 10^{2-4} cpds
- limited real data generation
- cheap
- **VERY** noisy
- scales to ultra-large libraries (10^{9-12})
- VS models have data requirements

Virtual Chemical Libraries



Hoffmann & Gastreich "The next level in chemical space navigation: going far beyond enumerable compound libraries." *Drug Discovery Today*, 2019, 24, 5, 1148-1156.

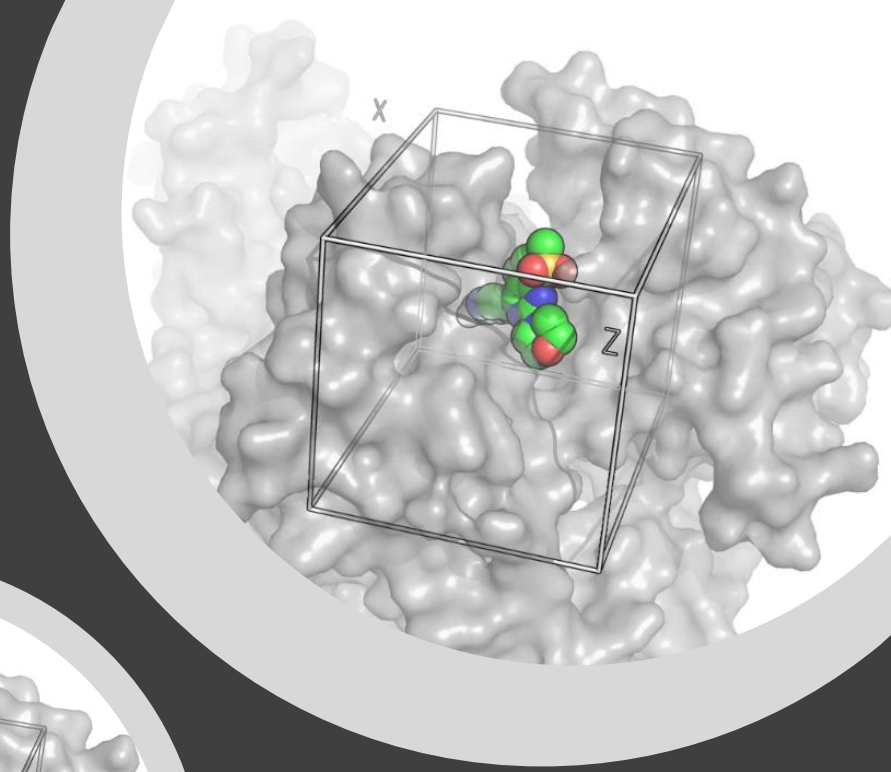
Walters P. "Virtual Chemical Libraries." *J. Med. Chem*. 2019, 62, 3, 1116-1124

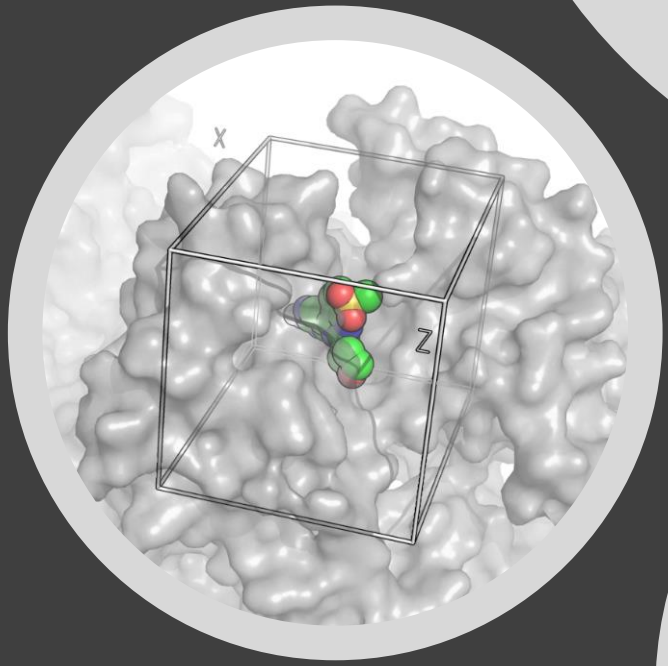
Structure-based virtual screening

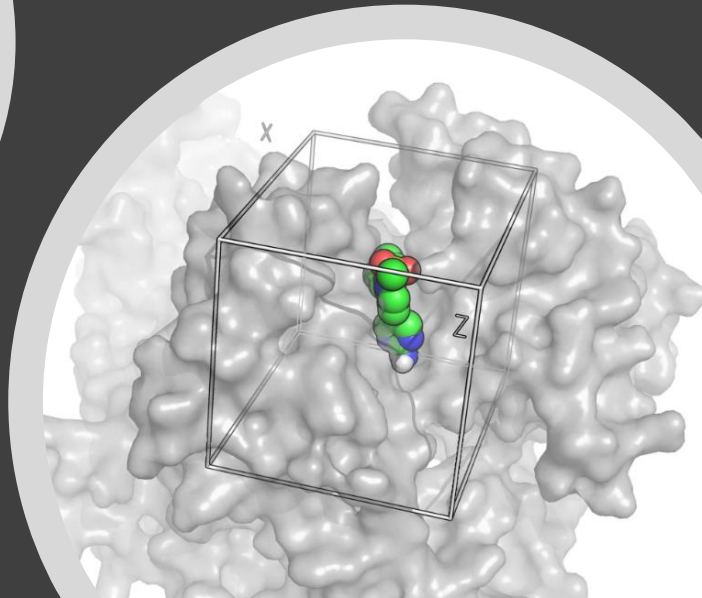
SBVS

What is docking (and co-folding?)

- Docking uses 3D molecular models to find best fit of molecule to active site of target.
- Search guided by a scoring function that evaluates favorability of each sampled configuration.
- Many docking programs are available.
- Docking score is crude estimate of binding favorability for a given compound.

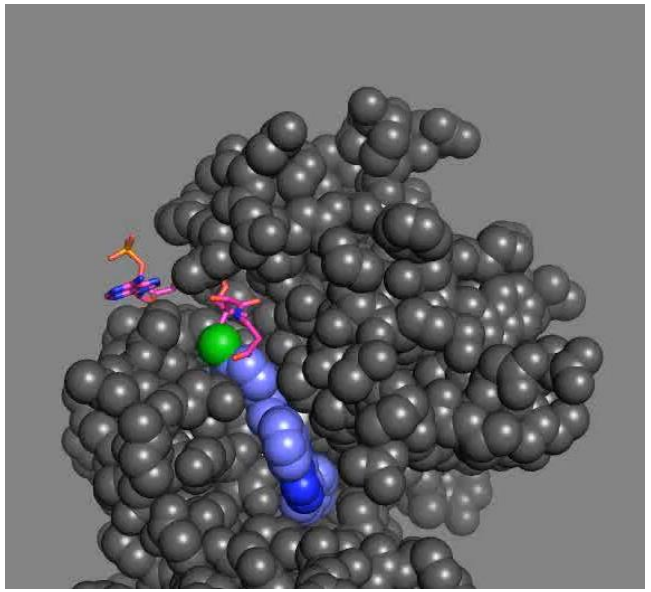






Structure-based virtual screening

Dock Compound Library

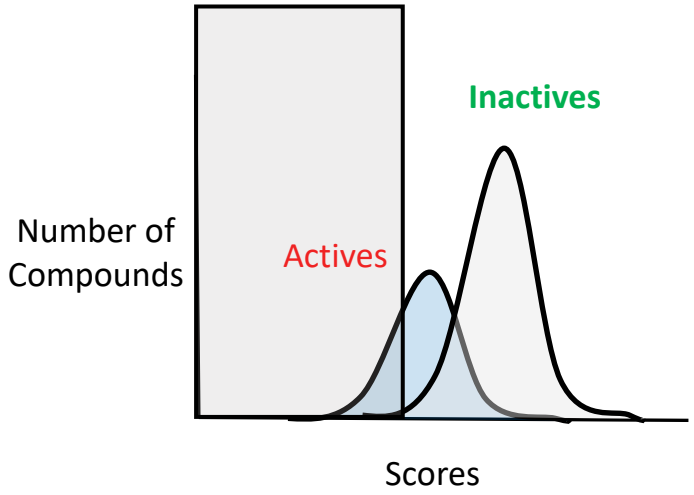


MOLID	SCORE
ZINC36206438	58.63
ZINC59310217	58.72
ZINC61596674	56.35
ZINC67458535	47.40
CHEMBL1221861	60.66
ZINC10123401	52.39
ZINC64526095	66.13
ZINC24002103	56.72
ZINC09612655	58.84
ZINC24002105	38.95
CHEMBL38532	74.19
ZINC40824467	50.10
ZINC59829723	58.29
ZINC37520295	44.78
ZINC49812309	38.01
ZINC14558020	53.31
CHEMBL472090	58.71
ZINC36207525	69.07
ZINC14010625	68.48
CHEMBL274782	63.97
ZINC63949457	55.35
ZINC39657146	48.74
ZINC23197109	58.72
ZINC25520953	63.14
ZINC09282496	43.71
ZINC60343267	62.18
ZINC58790750	62.53
CHEMBL400392	65.96
ZINC52096905	49.96
ZINC48922871	49.59
ZINC33058380	45.11
ZINC64684798	56.64
ZINC21076300	68.36
ZINC29461868	50.65
CHEMBL26183	58.56
ZINC61908006	66.40
ZINC15429053	54.10
CHEMBL323258	74.94
ZINC05091951	58.47
ZINC02759924	48.25
ZINC54596097	42.68
ZINC19899314	65.54
ZINC53113244	38.99
ZINC40947055	61.87
ZINC36611787	60.04
CHEMBL419085	65.96
ZINC35844701	58.57
ZINC01296699	39.07
ZINC39914438	49.68
ZINC00706129	48.34
ZINC34747432	52.55
ZINC43220997	47.45
ZINC37619890	54.49
ZINC15666896	55.50

Sort Compounds by Docking Scores

MOLID	SCORE
CHEMBL323258	74.94
CHEMBL38532	74.19
ZINC36207525	69.07
ZINC14010625	68.48
ZINC21076300	68.36
ZINC61908006	66.40
ZINC64526095	66.13
CHEMBL419085	65.96
CHEMBL400392	65.96
ZINC19899314	65.54
CHEMBL274782	63.97
ZINC25520953	63.14
ZINC58790750	62.53
ZINC60343267	62.18
ZINC40947055	61.87
CHEMBL1221861	60.66
ZINC36611787	60.04
ZINC09612655	58.84
ZINC59310217	58.72
ZINC23197109	58.72
CHEMBL472090	58.71
ZINC36206438	58.63
ZINC35844701	58.57
CHEMBL26183	58.56
ZINC05091951	58.47
ZINC59829723	58.29
ZINC24002103	56.72
ZINC64684798	56.64
ZINC61596674	56.35
ZINC15666896	55.50
ZINC63949457	55.35
ZINC37619890	54.49
ZINC15429053	54.10
ZINC14558020	53.31
ZINC34747432	52.55
ZINC10123401	52.39
ZINC29461868	50.65
ZINC40824467	50.10
ZINC52096905	49.96
ZINC39914438	49.68
ZINC48922871	49.59
ZINC39657146	48.74
ZINC00706129	48.34
ZINC02759924	48.25
ZINC43220997	47.45
ZINC67458535	47.40
ZINC33058380	45.11
ZINC37520295	44.78
ZINC09282496	43.71
ZINC54596097	42.68
ZINC01296699	39.07
ZINC53113244	38.99
ZINC24002105	38.95
ZINC49812309	38.01

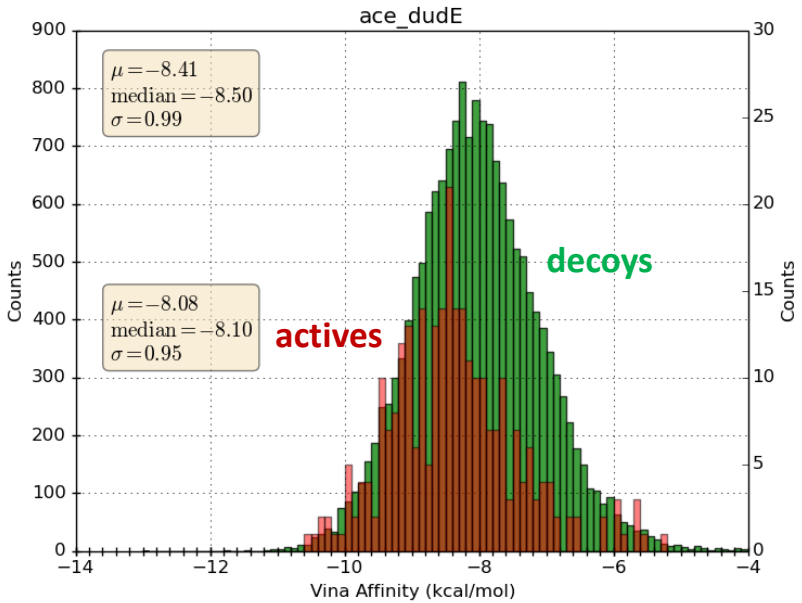
Score Distributions

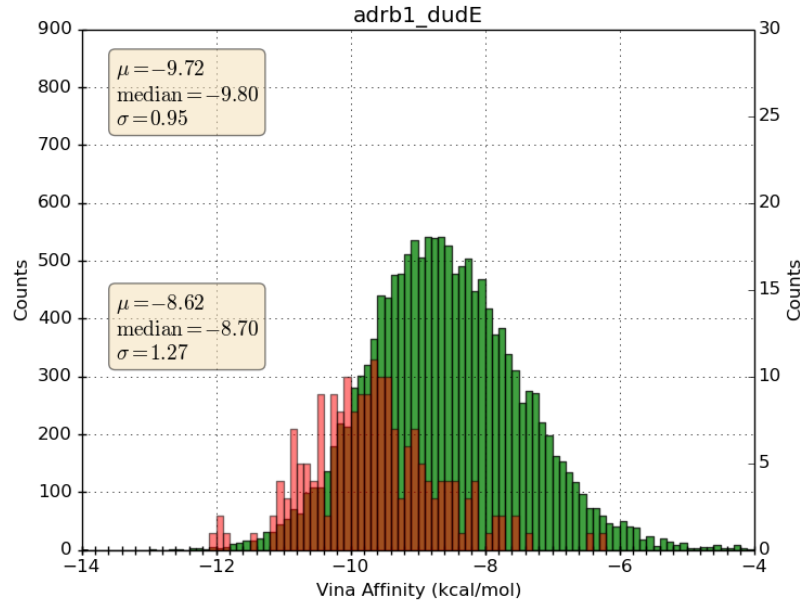


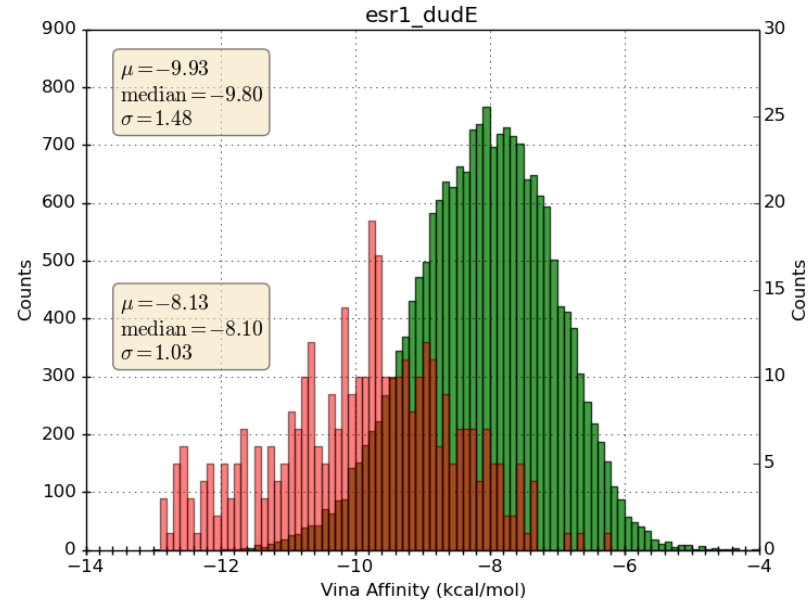
Examples of SBVS performance using docking program Vina on 6 benchmark targets from DUD-E

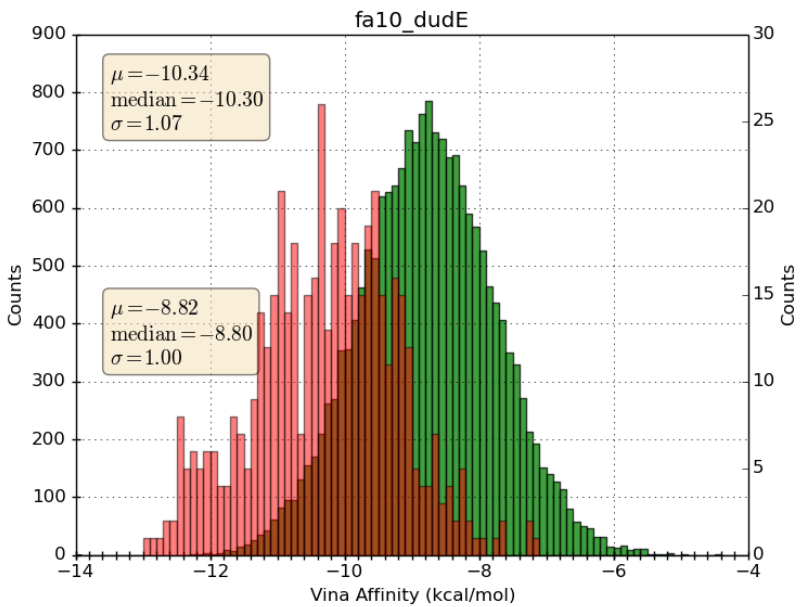
http://dude.docking.org/

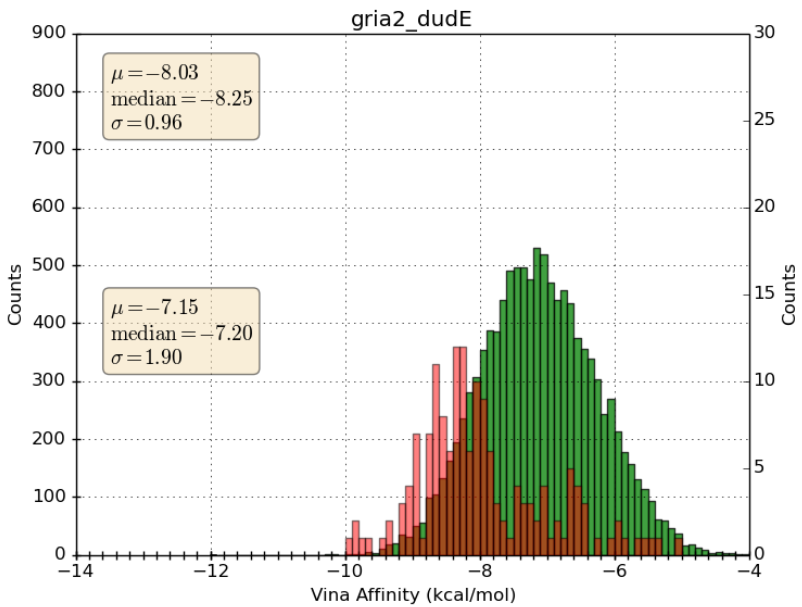
Red = score distribution for known actives
Green = score distribution for decoys (presumed inactives)

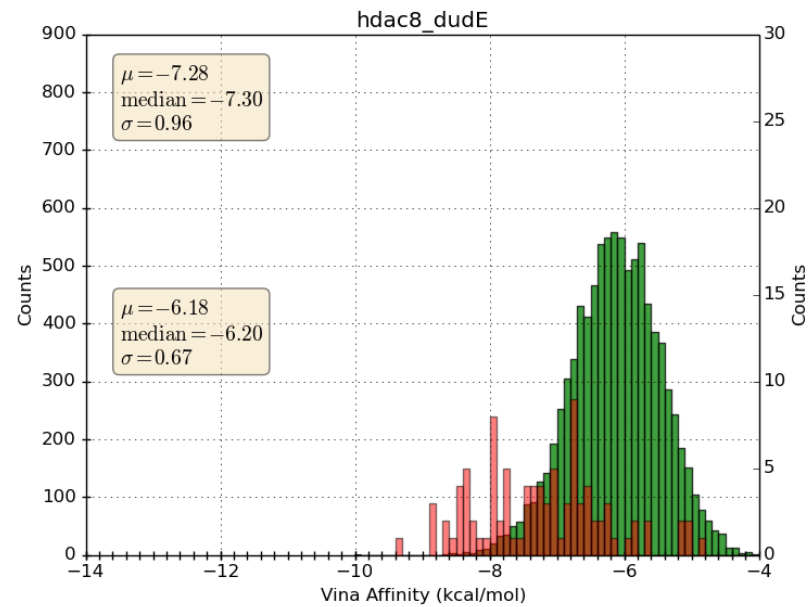






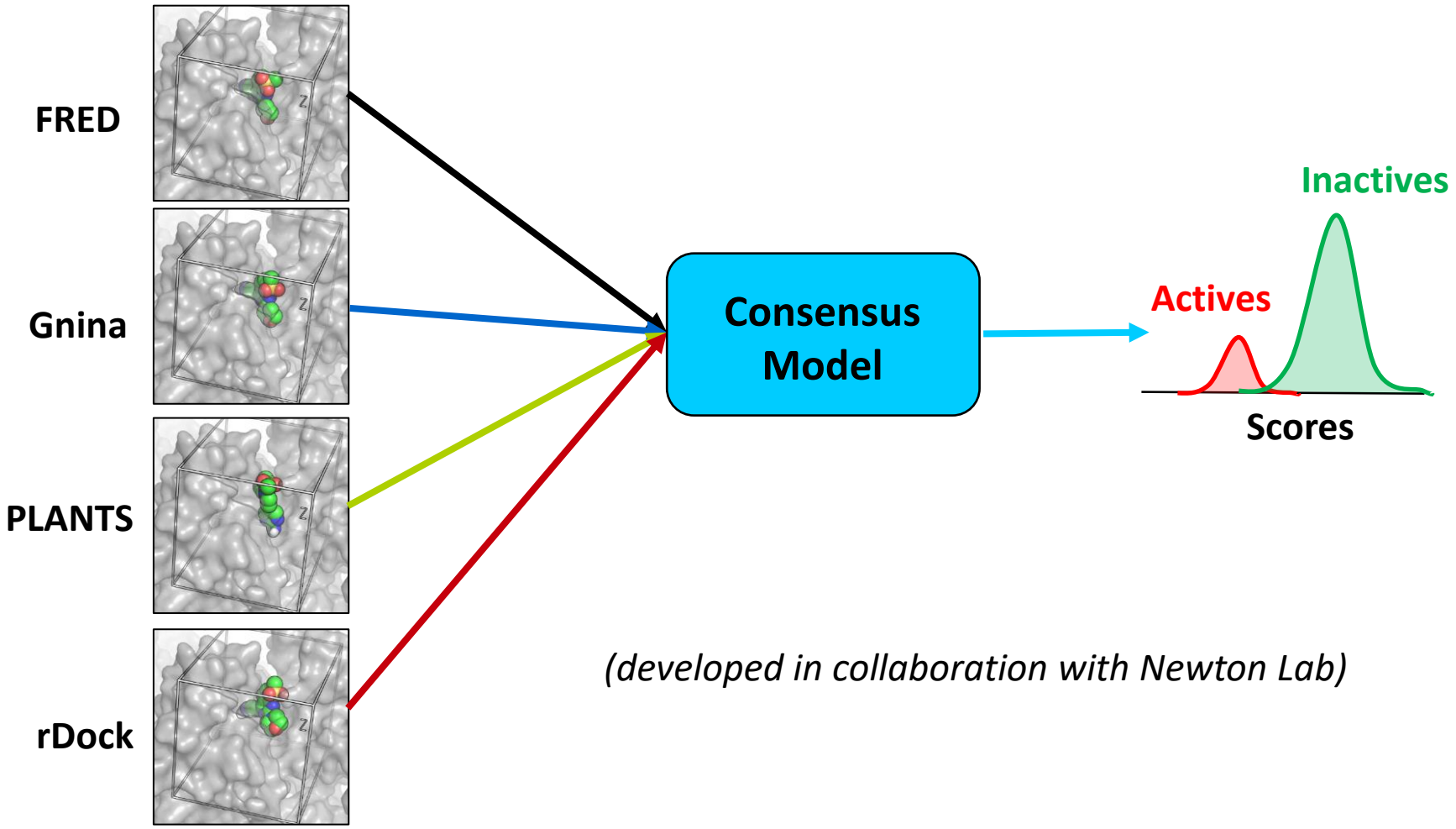






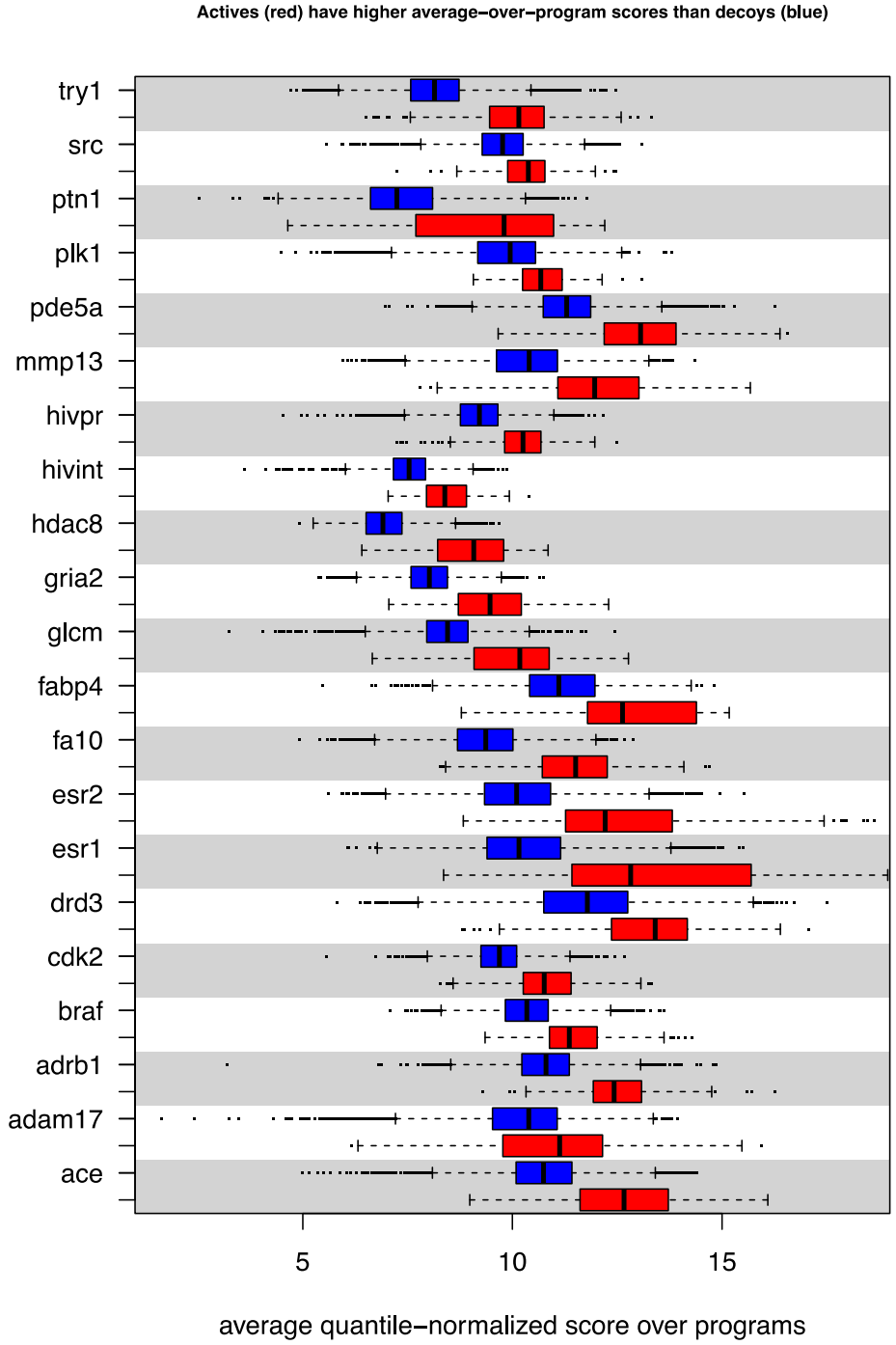
Consensus Scoring

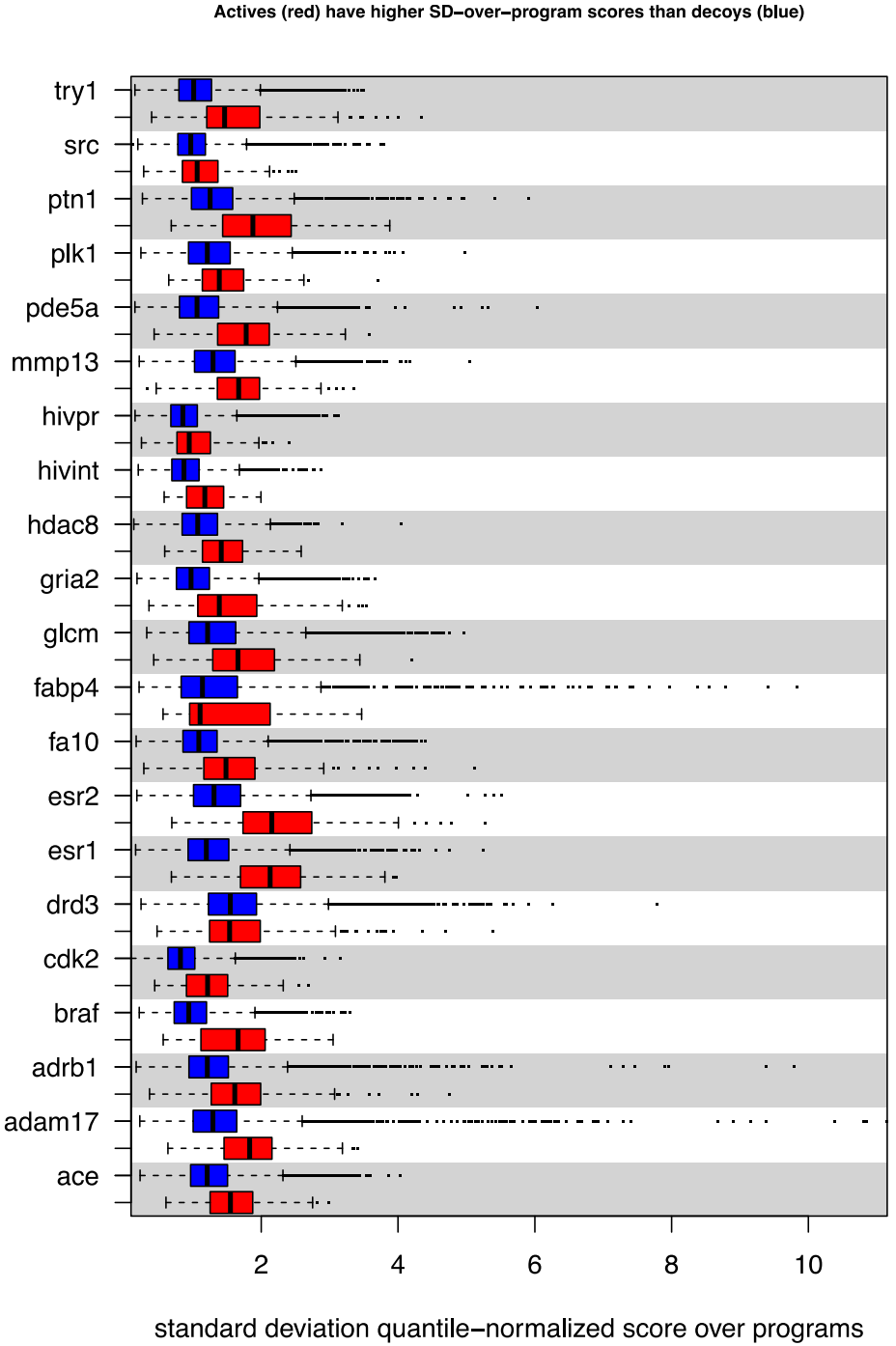
- No single program is reliable
- Use multiple docking programs
- Consensus scores are more reliable than those from any individual docking program.



(developed in collaboration with Newton Lab)

Ericksen et al., *J. Chem. Inf. Model*. **2017**, 57, 7, 1579-1590 DOI: 10.1021/acs.jcim.7b00153

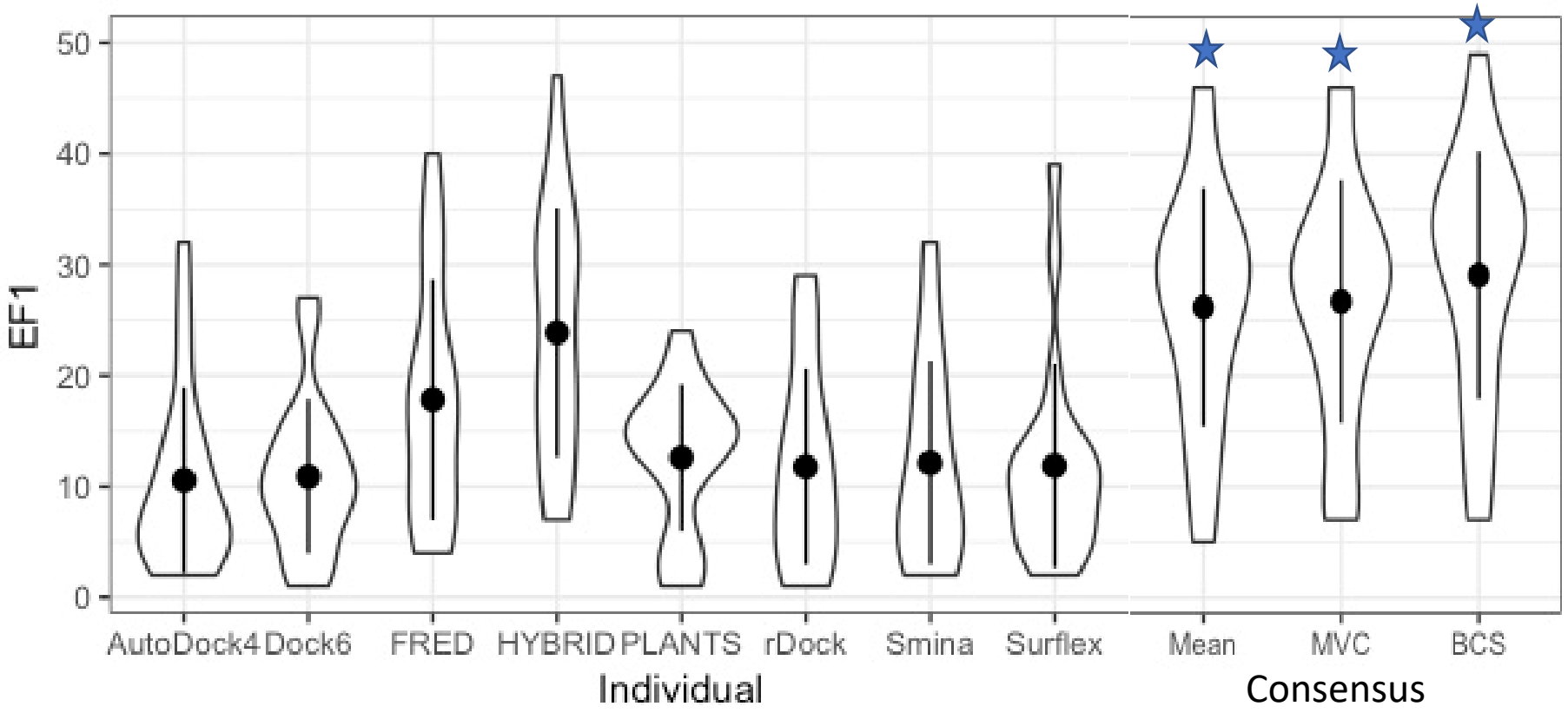




- Score distribution of actives (red) is shifted relative to inactives (blue).
- Interestingly, the standard deviation in scores was also higher for actives than for decoys

Ericksen et al., J. Chem. Inf. Model. **2017**, 57, 7, 1579-1590
DOI: 10.1021/acs.jcim.7b00153

Virtual Screening Performance on 21 benchmark targets from DUD-E



★ *P < 0.05*
Pairwise t-test

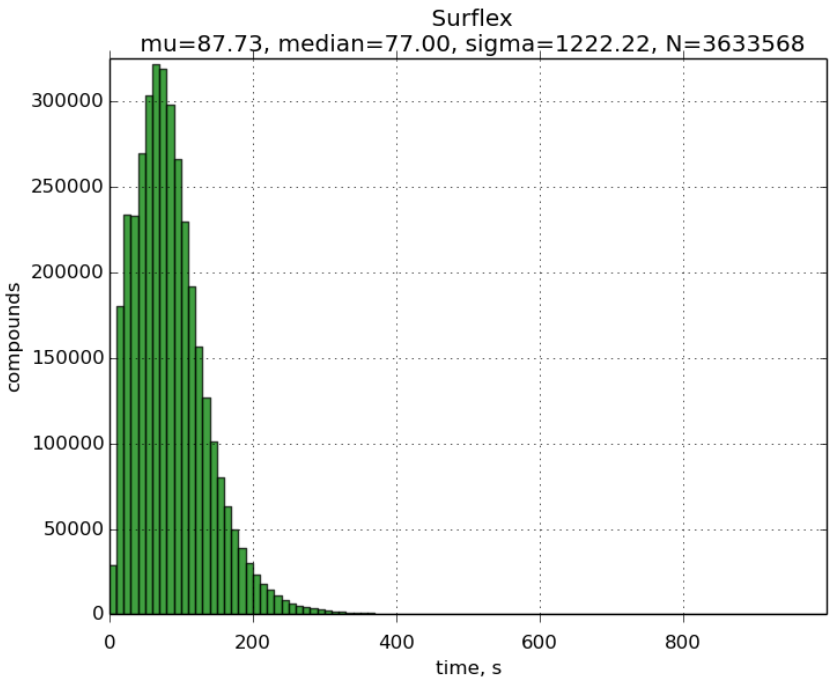
EF1 is a standard VS metric. It is the fold-enrichment for actives observed in the top 1% scoring cpds from a docked chemical library. EF1 = 1 indicates no enrichment or random.

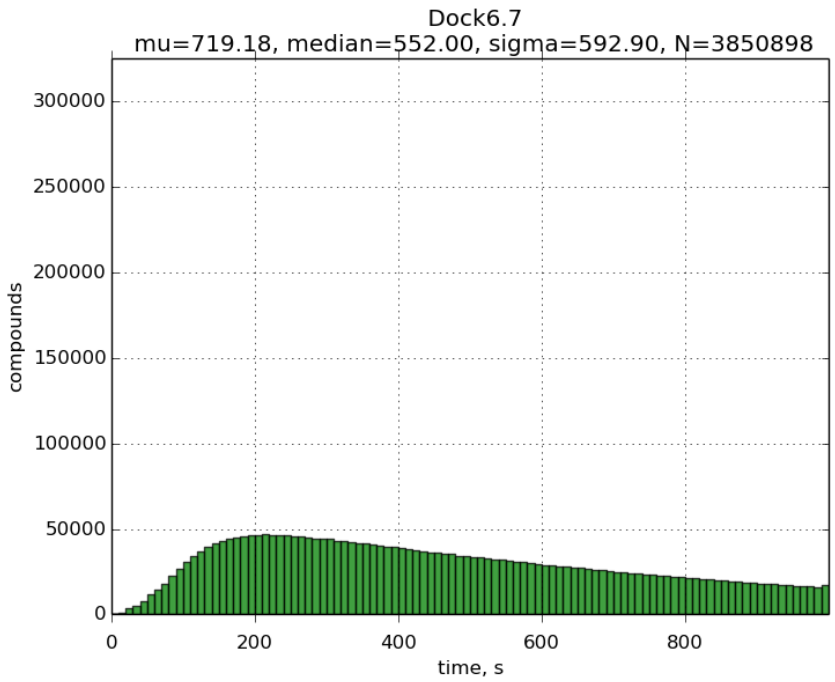
Target Class	Target
GPCR	ADRB1
GPCR	DRD3
Ion Channel	GRIA2
Kinase	BRAF
Kinase	CDK2
Kinase	PLK1
Kinase	SRC
Miscellaneous	FABP4
Receptor	ESR1
Receptor	ESR2
Other Enzymes	ACE
Other Enzymes	GLCM
Other Enzymes	HDAC8
Other Enzymes	HIVINT
Other Enzymes	PDE5A
Other Enzymes	PTN1
Protease	ADA17
Protease	FA10
Protease	HIVPR
Protease	MMP13
Protease	TRY1

Ericksen et al., *J. Chem. Inf. Model*. **2017**, 57, 7, 1579-1590 DOI: 10.1021/acs.jcim.7b00153

Docking Compute Expense

- Compute time for docking depends the search space, search quality, and complexity of the scoring function.
- To dock millions of compounds, we cut corners.
- Docking time varies between programs (~1 minute/compound).





(seconds)

Program	Time	Std. Dev.
AD4	435.6	197.1
Dock	719.2	592.9
Fred	15.6	5.7
Hybrid	9.3	2.9
Plants	43.4	20.5
rDock	49.3	26.7
Smina	250.1	172.8
Surflex	78.9	1159.6

Scaling SBVS on HTC resources

- Each docking run is independent--*pleasantly parallelizable*!
- Typical docking codes don't benefit from specialized hardware or multiple cores.
- Moved to docker containers for **rDock** and **Gnina**, use staging for **FRED**

- To maximize throughput:
    - Enable "`Flock`" and "`Glide`" to access more nodes.
    - Split compound library up into small chunks.
        - Number of compounds should run in ~2hr for a given docking program.
        - Chunk size varies from 5—500 compounds!
    - Dock each chunk on a single slot to scavenge ANY open slots. Compounds within each chunk are docked serially
    - ~~Checkpointing is enabled and a wrapper script is used to track the compounds completed in case job is evicted and migrates to another node.~~

```
.
├── ligs_1
│   └── ligs_1.mol2
├── ligs_2
│   └── ligs_2.mol2
├── ligs_3
│   └── ligs_3.mol2
├── ligs_4
│   └── ligs_4.mol2
├── run_gnina.sh
├── run_gnina.sub
└── shared
    └── 9n93_chimerax_H.mol2
```

run_gnina.sub

```
#; Specify the HTCondor Universe (vanilla is the default and is used
#; for almost all jobs), the desired name of the HTCondor log file,
#; and the desired name of the standard error and standard output file.
universe = docker

#; add a line to indicate which docker image we want to use
docker_image = gnina/gnina:latest

log = process.log
error = process.err
output = process.out

#; If your jobs are less than 4 hours long, "flock" them additionally to
#; other HTCondor pools on campus by uncommenting the below line:
+WantFlocking = true

#; If your jobs are less than ~2 hours long, "glide" them to the national
#; Open Science Grid (OSG) for access to even more computers and the
#; fastest overall throughput. Uncomment the below line:
+WantGlidein = true

#; Specify your executable (single binary or a script that runs several
#; commands) and arguments
executable = run_gnina.sh

#; Specify that HTCondor should transfer files to and from the
#; computer where each job runs.
should_transfer_files = YES
when_to_transfer_output = ON_EXIT_OR_EVICT

#; Set the submission directory for each job with the $(directory)
#; variable (set below in the queue statement).  Then transfer all
#; files in the shared directory, and from the submission directory
initialdir = $(directory)
transfer_input_files = ../shared/,./

#; Tell HTCondor what amount of compute resources
#; each job will need on the computer where it runs.
request_cpus = 1
request_memory = 4GB
request_disk = 6GB

#; Create a job for each "job" directory.
max_materialize = 5000
#; when using max_materialize, use: condor_submit -factory run.sub
queue directory matching ligs_*
```

run_gnina.sh

```
#!/bin/bash

ligs=`ls ligs_*.mol2`
ligname=`echo $ligs | cut -f1 -d"."`

gnina  --cpu 1 \
       --center_x 132.9 \
       --center_y 127.3 \
       --center_z 135.6 \
       --size_x 28.0 \
       --size_y 20.0 \
       --size_z 18.6 \
       -r 9n93_chimerax_H.mol2 \
       -l ${ligs} \
       -o out_${ligname}_9n93.sdf.gz \
       --log out_${ligname}_9n93.log \
       --exhaustiveness 16 \
       --seed 0 \
       --cnn_scoring rescore \
       --no_gpu
```

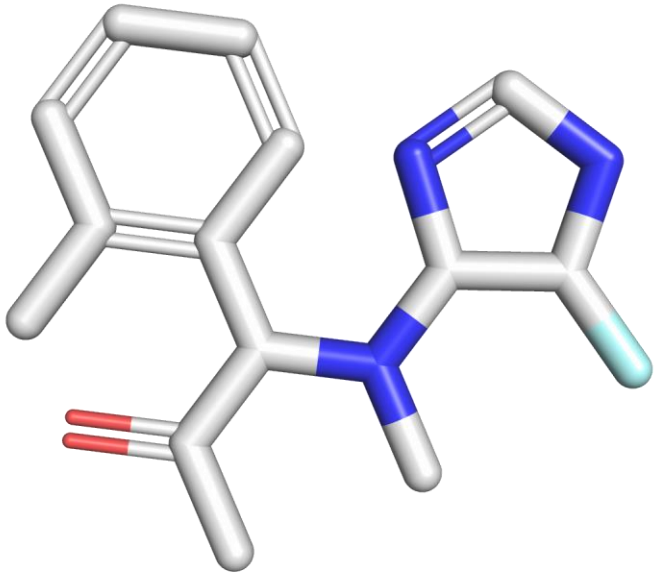
SBVS benefits from HTC

- Couldn’t see how docking-based VS works without proper fiddling/testing/validation!
- Examine performance over many targets
- Benchmarking of different docking programs
- Extensive docking parameter testing/validation
- Dock large compound sets
  - Routinely perform SBVS on libraries of 3-30 million cpds

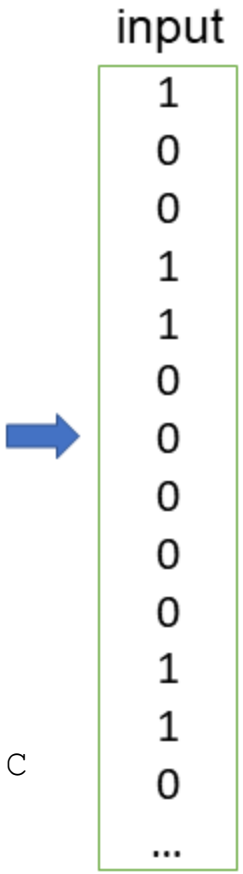
ligand-based virtual screening

LBVS

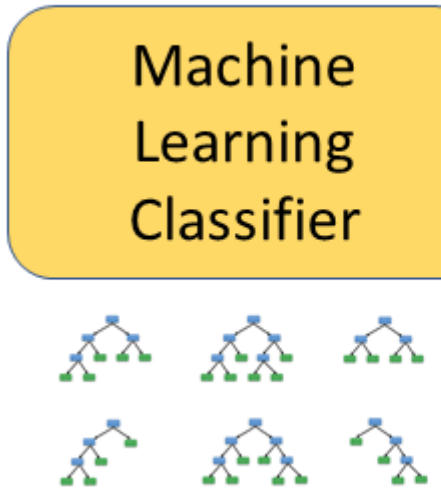
Ligand-Based Virtual Screening—a ML hit-finding model

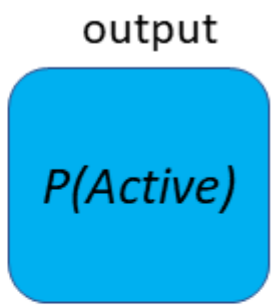


C1=CC(=C(C=C1)C(C)=O)N(C)C2C(NC=N2)F)C



Chemical Fingerprint





Binary classification

Gitter Lab: Liu, et al., "**Practical Model Selection for Prospective Virtual Screening.**" *J. Chem. Inf. Model.* 2019, 59, 1, 282–293. https://doi.org/10.1021/acs.jcim.8b00363

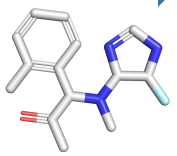
ML-Based VS on Ultra-Large Chemical Library

Ultra-Large Virtual Chemical Library

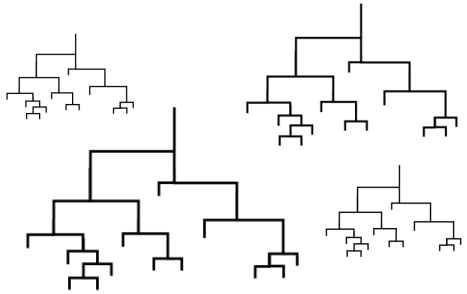
Enamine REAL database

1.1 billion potential molecules

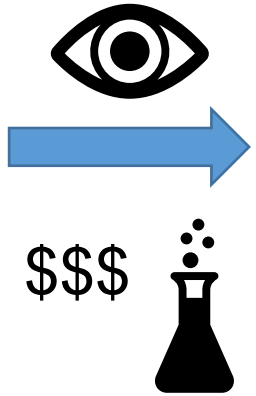
“synthesize-on-demand”



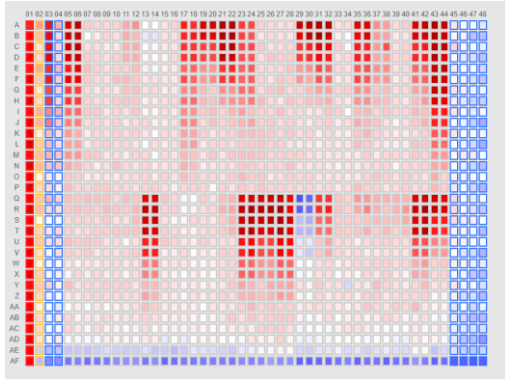
Random Forest Classifier



Prior HTS Training Data



Acquire/Test Molecules Prioritized by Classifier



Alnammi M. et al., “Evaluating scalable supervised learning for synthesize-on-demand chemical libraries.” ChemRxiv 2021.

VS on Ultra-Large Virtual Chemical Library

Train RF model on prior screening data (PriA-SSB interaction)

- LifeChem Diversity Sets 1-3: 75,000 cpds (primary and retest)
- LifeChem Diversity Set 4: 25,000 cpds (primary only)
- MLPCN (NIH probe set): 337,000 cpds (primary and retest)

Training Data: 427,000 cpds, number of actives: 554 (hit rate = 0.13%)

VS Procedure

- https://github.com/gitter-lab/pria-ams-enamine/tree/master
- Download Enamine REAL database 1.1 billion molecules (Oct 11, 2019)
- Split library up into 18 batches (each 60.3 million)
  - Average compute time of **3.24 ms per compound**
  - Mean run time per 60 million cpd batch = 53.2 hrs

https://enamine.net/compound-collections/real-compounds/real-database

Gitter Lab: Alnammi M. et al., "Scalable supervised learning for synthesize-on-demand chemical libraries." manuscript in prep

Dose-response testing of 68 compounds ordered from Enamine

IC50 (uM)	C 0.5 (% ne	C 1.0	C 2.1	C 4.1	C 8.2	C 16	C 33	C 66	Active
0.5	93.2	40.8	21.0	11.6	8.1	6.6	5.9	8.6	1
2.1	78.2	66.2	45.9	25.2	10.3	7.8	7.1	6.0	1
0.5	116.4	69.3	50.5	24.2	11.4	8.8	7.1	10.3	1
3.4	79.7	74.1	63.1	33.9	9.9	6.0	6.2	5.3	1
4.4	85.3	78.5	72.0	47.9	11.8	4.5	3.5	3.4	1
4.8	72.8	63.2	54.6	43.7	33.6	19.4	10.5	8.3	1
4.9	85.7	77.9	70.4	54.4	19.6	8.0	5.6	5.7	1
6.1	80.1	67.9	67.8	53.2	33.5	15.1	9.0	8.6	1
6.8	88.1	83.0	81.0	68.8	35.0	10.4	5.5	4.5	1
9.1	85.0	81.5	81.0	73.0	50.9	14.3	6.5	6.2	1
3.9	119.8	86.5	82.0	68.2	47.7	15.7	7.3	9.8	1
2.4	126.3	91.8	81.4	72.5	47.3	19.4	7.6	8.5	1
10.8	86.8	84.3	79.5	76.0	58.9	24.2	8.8	6.2	1
0.5	132.0	91.3	82.4	73.1	56.0	24.0	9.9	9.5	1
12.4	85.5	85.9	85.4	79.8	63.8	30.9	8.0	5.0	1
48.2	81.6	75.7	67.9	60.5	55.0	45.8	30.8	25.8	1
66.0	121.5	88.3	82.6	75.7	63.1	37.5	9.1	7.2	1
12.6	87.7	87.0	82.8	83.9	71.1	28.1	8.0	6.1	1
12.1	89.2	89.7	86.1	80.1	66.8	30.2	10.3	6.7	1
11.5	91.4	87.2	87.5	84.5	67.7	26.5	10.9	6.3	1
15.5	83.3	88.0	77.3	74.0	62.2	41.1	24.7	12.3	1
35.4	123.9	94.4	86.1	81.9	68.7	40.8	9.4	8.4	1
13.9	87.7	86.1	87.4	85.7	72.9	38.0	11.4	9.3	1
21.0	86.6	83.3	82.5	75.9	67.8	47.8	24.9	9.2	1
15.7	87.8	89.6	89.8	83.9	70.5	43.8	15.8	5.7	1
7.0	80.0	78.9	78.9	71.7	53.7	44.8	39.8	36.9	1
66.0	125.3	92.1	86.0	86.3	74.4	49.5	15.4	10.8	1
22.2	90.0	90.6	92.1	86.8	76.4	62.5	22.7	7.2	1
37.8	88.5	87.6	87.0	80.1	77.6	62.3	36.5	11.0	1
24.9	84.6	87.5	88.0	89.1	80.0	65.2	29.9	8.9	1
66.0	140.0	94.7	97.5	91.0	83.1	62.3	25.1	10.1	1
20.2	86.5	85.7	83.3	80.3	80.3	68.4	50.3	43.6	0
66.0	136.0	100.0	89.3	86.0	86.7	74.1	49.7	28.3	1
66.0	83.7	86.8	87.7	82.8	76.5	63.1	60.3	65.1	0
66.0	86.9	87.8	85.9	86.7	78.5	78.8	71.7	69.4	0
7.3	131.5	93.7	90.9	90.4	90.5	82.4	70.5	49.2	0
66.0	122.0	95.1	90.0	84.9	88.9	84.5	74.2	63.6	0
66.0	83.3	83.6	80.6	82.6	82.0	86.1	84.0	86.0	0
66.0	84.2	82.1	84.5	83.4	83.3	81.6	84.1	87.9	0
66.0	134.3	95.6	87.6	85.4	85.2	84.7	77.4	72.8	0
66.0	134.5	97.4	87.2	83.1	84.6	86.1	77.9	74.2	0
66.0	136.5	93.6	88.8	89.0	87.8	89.4	78.2	63.8	0
66.0	85.7	86.8	83.5	80.2	87.7	82.1	84.7	87.2	0
66.0	89.4	88.1	89.9	89.3	81.7	82.2	80.6	80.9	0
66.0	87.3	84.5	88.6	83.2	86.9	81.6	85.0	88.7	0
66.0	86.8	85.6	84.8	86.7	86.2	83.9	88.4	85.0	0
66.0	134.3	100.4	87.4	88.0	89.0	87.8	77.2	71.0	0
66.0	89.2	88.1	87.4	89.4	88.2	84.2	82.2	82.8	0
66.0	95.1	87.1	84.9	87.5	87.3	88.0	86.8	81.4	0
66.0	83.7	85.4	88.7	88.1	86.9	86.8	85.6	84.4	0
66.0	131.0	93.0	91.7	84.5	87.9	87.0	82.7	79.7	0
66.0	129.3	95.0	89.2	86.5	90.2	87.5	80.7	78.1	0
66.0	88.1	88.0	83.6	85.4	83.6	89.5	86.7	90.7	0
66.0	136.3	93.7	92.4	86.6	91.3	81.6	85.5	78.3	0
66.0	138.0	96.4	92.7	85.1	89.7	82.9	84.5	78.9	0
66.0	85.0	88.3	91.9	88.0	87.6	88.7	84.8	86.7	0
66.0	87.8	88.8	90.8	88.7	88.9	84.7	89.5	84.7	0
66.0	88.2	88.0	88.1	89.4	88.9	91.0	90.2	87.0	0
66.0	130.5	94.1	93.9	93.5	93.2	86.5	81.2	81.4	0
66.0	119.8	91.6	93.1	90.0	88.0	87.7	90.8	84.4	0
66.0	126.3	96.0	92.6	91.1	92.1	87.8	85.4	81.5	0
66.0	89.3	90.8	91.0	89.7	86.5	88.6	88.2	91.7	0
66.0	141.3	99.5	92.2	89.1	91.5	86.7	87.8	82.3	0
66.0	131.5	95.4	94.9	91.6	95.0	88.0	87.6	85.4	0
66.0	125.3	94.6	94.9	89.9	90.1	93.7	88.5	87.4	0
66.0	133.0	96.7	92.9	91.7	93.3	91.3	91.6	97.2	0
66.0	85.7	90.2	92.9	92.2	89.1	92.9	96.6	101.8	0
66.0	140.8	101.8	93.2	93.1	97.6	108.8	125.5	170.5	0

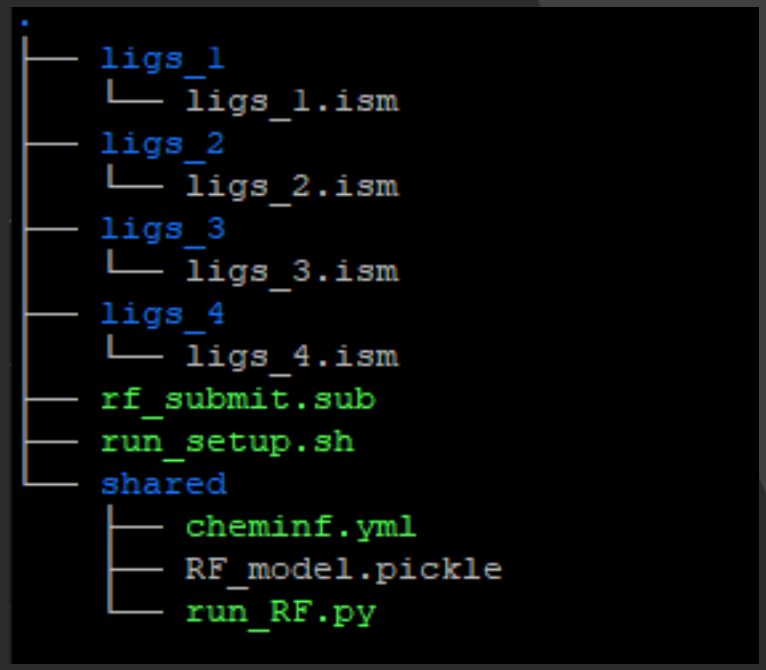
Running LBVS on HTC

- split very large library in chunks of SMILES (1 line → 1 compound) a few million lines per chunk (ligs_X.ism)

- **`rf_submit.sub`** copies folders with chunks of SMILES and contents of shared folder to execute nodes

- submit file runs specific executable--BASH script **`run_setup.sh`** which does the following:
  - downloads Anaconda (wget/curl) and runs Anaconda binary installer
  - runs conda on **`cheminf.yml`** to setup python cheminformatics environment (scikit-learn and rdkit), activates environment
  - runs python script **`run_RF.py`** which does the following:
    - uses rdkit module to convert SMILES to fingerprints (numpy array of input features)
    - uses scikit-learn module to load random forest classifier (**`RF_model.pickle`**) and predicts on fingerprints

- returns output CSV containing molecule IDs and classification results.



`ligs_1.ism`

```
Cn1c(c2ccc(cc2n1)Br)C(=O)NC3CC4CCC(C3)N4S(=O)(=O)C SMSSF-0616774
CS(=O)(=O)N1C2CCC1CC(C2)NC(=O)c3ccc(cc3)c4ccsc4 SMSSF-0616773
CS(=O)(=O)N1C2CCC1CC(C2)NC(=O)COc3ccccc3F SMSSF-0616772
CS(=O)(=O)N1C2CCC1CC(C2)NC(=O)C3CC3 SMSSF-0616771
c1ccc(cc1)c2cn(nn2)C3CN(C3)S(=O)(=O)N4CCCC4 SMSSF-0616770
Cc1c(ncn(c1=O)CC(=O)N2CC(C2)n3cc(nn3)c4ccccc4)C SMSSF-0616769
c1ccc(cc1)c2cn(nn2)C3CN(C3)C(=O)Cc4c5ccccc5on4 SMSSF-0616768
Cc1c(snn1)C(=O)N2CC(C2)n3cc(nn3)c4ccccc4 SMSSF-0616767
COc1ccc(c(c1)C(=O)N2CC[C@@H](C2)n3cc(nn3)c4ccccc4)Br SMSSF-0616766
Cn1cc(nc1)S(=O)(=O)N2CCN(CC2)c3cc(ncn3)n4cncn4 SMSSF-0616765
Cn1cc(cn1)S(=O)(=O)Nc2ccccc2c3csc(n3)c4ccccc4F SMSSF-0616764
Cc1nc(cs1)c2ccccc2NC(=O)Nc3ccccc3Cl SMSSF-0616763
```

rf_submit.sub

```
#; Specify the HTCondor Universe
universe = vanilla

log = process.log
error = process.err
output = process.out

#; If your jobs are less than 4 hours long, "flock" them additionally to
#; other HTCondor pools on campus by uncommenting the below line:
+WantFlocking = true

#; If your jobs are less than ~2 hours long, "glide" them to the national
#; Open Science Grid (OSG) for access to even more computers and the
#; fastest overall throughput. Uncomment the below line:
+WantGlidein = true

#; Specify your executable (single binary or a script that runs several
#;  commands) and arguments
executable = run_setup.sh
#;arguments = arguments to your script go here

#; Specify that HTCondor should transfer files to and from the
#;computer where each job runs.
should_transfer_files = YES
when_to_transfer_output = ON_EXIT_OR_EVICT

#; Set the submission directory for each job with the $(directory)
#; variable (set below in the queue statement).  Then transfer all
#; files in the shared directory, and from the submission directory
initialdir = $(directory)
transfer_input_files = ../shared/,./

#; Tell HTCondor what amount of compute resources
#; each job will need on the computer where it runs.
request_cpus = 1
request_memory = 4GB
request_disk = 6GB

#; Create a job for each "job" directory.
max_materialize = 5000
#; when using max_materialize, use: condor_submit -factory run.sub
queue directory matching chunk_*
```

run_setup.sh

```
#!/bin/bash
#install anaconda
export HOME=$_CONDOR_JOB_IWD
wget -q --retry-connrefused --waitretry=10 https://repo.continuu
m.io/archive/Anaconda3-5.3.0-Linux-x86_64.sh
chmod 777
./Anaconda3-5.3.0-Linux-x86_64.sh -b -p ./anaconda > /dev/null
export PATH=$PWD/anaconda/bin:$PATH
echo 'Done installing anaconda'
chmod 777 *

# setup rdkit python environment
conda env create -f cheminf.yml > /dev/null
source activate cheminf
echo 'Done installing environment'

# run python script
python run_RF.py
```

cheminf.yml

```
name: cheminf
channels:
- bioconda
- rdkit
- conda-forge
- default
dependencies:
- python=3.6
- setuptools
- rdkit=2018.03.4.0
- numpy=1.15.3
- pandas=0.23.4
- matplotlib=2.2.3
- scikit-learn=0.20.0
~
```

run_RF.py

```python
import pandas as pd
import numpy as np
import pathlib
import gzip
import time
import json
from rdkit import Chem
from rdkit.Chem import AllChem
from rdkit.Chem.SaltRemover import SaltRemover
from rdkit.Chem.FilterCatalog import *

import sys
sys.path.insert(0,'../../src/')
from random_forest_classification import RandomForestClassification

if __name__ == '__main__':
    start_time = time.time()

    # load random forest model
    with open('./139.json', 'r') as f:
        task_conf = json.load(f)
    rf_model = RandomForestClassification(conf=task_conf)
    rf_model = rf_model.load_model(rf_model_file)
    pathlib.Path(output_csv_file).parent.mkdir(parents=True, exist_ok=True)

    # process real db file in chunks, generating features, and predicting using random forest model.
    saltRemover = SaltRemover(defnFilename='../../datasets/raw/Salts.txt')
    FP_radius, FP_size = 2, 1024
    n_cpds = 0
    data_df = pd.read_csv(real_db_file, chunksize=chunksize,
                          usecols=[0], header=None, skiprows=1+process_id*instances_per_process,
                          nrows=instances_per_process, delimiter='\t')
    with open(output_csv_file, 'w') as outputfile:
        outputfile.write('smiles,rf_preds\n')
        for chunk_idx, chunk_df in enumerate(data_df):
            n_cpds += chunk_df.shape[0]
            chunk_smiles = chunk_df[0]
            rdkit_mols = chunk_smiles.astype(str).apply((lambda x: Chem.MolFromSmiles(x)))
            rdkit_mols = rdkit_mols.apply((lambda x: saltRemover.StripMol(x)))
            X = rdkit_mols.apply((lambda x: AllChem.GetMorganFingerprintAsBitVect(x,
                                                                                  radius=FP_radius,
                                                                                  nBits=FP_size).ToBitString()))
            X = np.vstack([np.fromstring(x, 'u1') - ord('0') for x in X]).astype(float) # this is from: https://stackoverflow.com/a/29091970
            chunk_preds = rf_model.predict_proba(X)[:,1]

            for smile, pred in zip(chunk_smiles.tolist(), chunk_preds):
                outputfile.write('{},{}\n'.format(smile, pred))

    end_time = time.time()
    total_time_minutes = (end_time-start_time)/60.0
    print('Total time: {} minutes to process {} cpds.'.format(total_time_minutes, n_cpds))
```

Conclusions

HTC is a fabulous resource for VS

Effective VS requires rapid cycles of development, testing, validation. **HTC enables this!**

HTC allows VS to scale to new ultra-large virtual chemical libraries

Acknowledgments

- CHTC, CHTC Facilitators:

  Miron Livny, Christina Koch, Lauren Michael, Amber Lim, Danny Morales, Rachel Lombardi, Jason Patton, Jessica Vera, …

- Funding

  Tony Gitter & Michael Newton “A Machine Learning Platform for Adaptive Chemical Screening.” 1R01GM135631-01A1

  Manish Patankar -- I01BX005627 (VA)

  UWCCC Shared Resources -- P30 CA014520

- Computational Chemists

  Ken Satyshur, Scott Wildman, Shengchao Liu, Moayad Alnammi, Yifan Deng, Haozhen Wu, Huikun Zhang, Peng Yu, Ryan Kassab

CENTER FOR HIGH THROUGHPUT COMPUTING

HTCondor High Throughput Computing

NIH National Institutes of Health *Turning Discovery Into Health*

VA U.S. Department of Veterans Affairs

Carbone Cancer Center UNIVERSITY OF WISCONSIN SCHOOL OF MEDICINE AND PUBLIC HEALTH

OpenEye SCIENTIFIC

Open Science Grid

DDC-SMSF

NSF

EXTRAS

Small Molecule Screening Facility (SMSF)

Scientific Highlights: collaboration with Gitter Lab (Morgridge) and Barzilay Lab (MIT/BoltzLab) -- paper in review

Announcing Boltz Lab and our first agents

January 8th, 2026 The Boltz Team

Over the past couple of months, we have been testing the agent with many academics and industry groups against a wide variety of targets. We will unveil more experimental results in various upcoming publications, but we can already share a first successful validation of the agent against TYK2—a target that was excluded from the training set of the model—conducted in collaboration with Song Guo, Spencer S. Ericksen, and Anthony Gitter from the University of Wisconsin-Madison. We ordered 29 top compounds selected by the agent from Enamine, of which 23 were successfully synthesized. Of those 23, we saw 3 actives, 8 weakly-actives, and 12 inactives, with the strongest binder having a measured **IC50 of 240 nM**.

Synonyms	IC50 uM	Hit Assignment	3.0E-09	1.0E-08	3.0E-08	1.0E-07	3.0E-07	1.0E-06	3.0E-06	1.0E-05
TC JL 37 (+ ref)	ND		25.7	60.3	80.1	91.4	93.6	96.1	98.6	101.0
Baricitinib (+ ref)	ND		18.1	51.3	76.1	89.9	93.3	95.9	99.2	103.2
Z1716625760	0.28	Active	-5.9	-3.7	3.8	25.1	51.2	65.0	82.8	94.9
Z2094189446	1.10	Active	3.0	-3.7	4.7	3.6	29.6	41.2	84.3	94.3
Z9807430340	1.71	Active	-0.7	-0.2	2.4	4.1	20.5	27.6	85.4	110.3
Z1039070116	ND	Active	-6.5	-6.5	-2.7	2.3	12.1	33.2	58.7	92.4
Z365689922	1.24	Active	6.3	2.8	2.2	8.0	15.4	27.5	47.5	56.5
Z4254670628	ND	Active	2.0	2.4	4.9	6.2	15.6	22.5	40.4	73.4
Z9807430358			0.2	0.5	5.5	4.0	22.5	24.9	55.1	33.2
Z9807430404	ND	Weak	0.8	0.6	6.2	5.3	21.3	20.7	40.0	44.8
Z2711330244			7.2	-1.2	7.2	6.8	11.1	16.4	30.7	70.0
Z2370251987			3.9	-2.1	5.1	4.6	19.4	21.8	28.8	57.8
Z9783868792	2.14	Active	-4.2	-4.7	-4.0	-2.1	4.3	17.0	56.2	82.1
Z9783868932	ND	Weak	-0.7	-0.7	0.3	1.5	9.6	15.4	43.2	69.0
Z9783868812	3.96	Active	-0.4	-2.1	-2.3	-2.1	3.1	11.7	45.2	75.1
Z9807430356	ND	Weak	-0.5	-0.8	1.1	1.2	10.2	14.9	34.4	54.5
Z9783868912	2.92	Active	-0.9	-0.8	0.1	0.7	6.8	14.1	36.6	57.1
Z9807430509	ND	Weak	2.2	2.1	4.6	6.4	10.4	14.7	26.6	38.0
Z1958652921	5.31	Active	-2.1	-3.8	-2.4	-2.7	4.9	16.1	38.1	68.1
Z5334523995	3.52	Active	-5.6	-6.6	-1.9	-0.9	1.0	15.1	41.1	70.4
Z199467708			1.1	0.7	0.2	0.8	6.8	13.9	33.8	33.4
Z224574650			3.7	-2.0	4.3	4.4	16.2	12.5	16.9	27.8
Z5332913906	ND	weak	-3.9	-4.0	-4.6	-3.7	2.1	12.9	31.5	63.6
Z1855774939	6.00	Active	-5.9	-6.6	-5.8	-5.1	2.5	12.4	35.6	67.1
Z643780490	ND	Weak	-1.3	-2.1	-2.0	-0.6	2.3	6.1	26.0	68.1
Z991902156	ND	Weak	-4.2	-4.1	0.4	0.3	-0.4	13.1	25.0	53.3
Z9807430413	ND	Weak	0.3	0.8	2.7	2.8	5.7	9.9	18.3	29.1
Z1942570678	ND	Weak	-3.8	-4.2	-1.0	-1.5	4.1	5.2	28.4	45.3
Z5217360448	ND	weak	-2.3	-2.4	-4.2	-3.0	1.2	7.2	21.9	52.8
Z2765340784			5.1	1.1	5.7	8.1	4.3	7.6	5.3	5.4
Z9783868832	5.25	Weak	2.9	1.6	0.4	-1.9	-0.1	-1.3	6.6	56.0
Z2975808667			5.4	2.0	3.2	1.9	1.9	4.1	6.0	22.2
Z221248194			1.3	1.0	4.1	2.4	1.8	0.8	8.8	23.4
Z1816002356	5.37	Weak	-6.7	-7.6	-7.6	-8.2	-4.2	3.3	24.8	59.7
Z1958348882	4.71	Weak	-5.3	-6.1	-6.1	-6.9	-3.1	5.0	22.6	49.2
Z2155079367			-0.1	-0.9	0.7	-0.7	3.7	3.3	14.5	14.7
Z9842664028	ND	Weak	0.9	0.3	0.4	0.6	2.0	3.2	8.3	23.9
Z9807430482	ND	Weak	-0.2	-0.4	0.3	0.0	1.5	3.9	9.6	20.6
Z730152598			-4.1	-4.6	-1.4	-3.2	5.2	3.0	17.2	16.9
Z4425235337			3.6	2.4	0.2	-0.2	-1.4	0.1	8.1	19.3
Z9807430446	ND	Weak	-0.4	-0.6	-0.7	-0.4	1.0	3.0	6.5	18.4
Z9783868872			-0.5	-0.3	-0.1	-0.4	1.1	0.9	8.3	10.3
Z9807430463	ND	Weak	-0.3	-0.4	-0.4	-0.6	1.0	1.5	5.0	16.5
Z5332027453	ND	weak	-3.4	-4.3	-4.6	-5.0	-1.6	0.8	10.4	37.8
Z9807430432	ND	Weak	-0.5	-0.8	-0.8	-0.8	0.5	1.6	5.3	15.1
Z99494835			-5.3	-2.9	-2.7	-3.7	0.4	0.3	11.3	14.0
Z2972769192			2.9	-2.8	-1.3	-2.3	-0.4	-1.0	5.4	12.8
Z56977048			4.8	3.3	1.3	0.2	-1.3	-1.7	0.2	0.0
Z3359593295			3.0	3.7	0.3	0.1	-0.9	-0.1	-0.5	0.3
Z1461252371			-0.9	-2.1	-0.2	-0.9	-1.2	0.3	0.2	1.8
Z285715042			-0.6	-1.6	-1.7	-0.2	-1.1	-0.9	-1.4	0.9
Z1119711320	ND	Weak	-5.8	-6.8	-4.8	-5.8	-5.0	-4.8	-1.5	40.9
Z1648064716			-6.1	-5.9	-7.5	-6.4	-4.6	-0.7	4.9	16.7
Z2054878068			-1.4	-7.1	-5.7	-6.7	-0.4	-1.6	4.2	5.0
Z1396689146			-3.4	-4.8	-2.7	-3.9	-3.0	-3.5	-0.6	5.9
deucravacitinib (- ref)	ND		-7.5	-7.8	-6.1	-6.6	-6.3	-6.2	-3.0	7.8

https://boltz.bio/boltzlab

Computational Services of the Drug Development Core

- **Hit Finding**
  - Virtual Screening
    - Structure-Based
    - Ligand-Based
  - Data Mining Public Bioassay Data for Target-Focused Libraries
- **Screening follow-up**
  - —HTS Data Analysis
  - —In silico PK/ADMET
  - —Similarity Searches
  - —Chemical Clustering
  - —Property Filters
  - —Molecule Purchasing
- **Lead Optimization**
  - —QSAR (ML)
  - —Free Energy Calcs (MD)
  - —Co-folding (Boltz-2)

School of Medicine and Public Health
UNIVERSITY OF WISCONSIN–MADISON

Computational Services of the Drug Development Core

Hit Finding

- Virtual Screening (VS)
  - Structure-Based (SBVS)
  - Ligand-Based (LBVS)
- Public Data Mining for Target-Focused Libraries
- Model-Guided Iterative Screening
- Probe Set Selection

Hits Triage

- HTS Data Analysis
- Similarity Searches
- Chemical Clustering
- Property Prediction
- Property Filters
- Hit Expansion
- Molecule Purchasing

Lead Optimization

- ML-based QSAR
- Molecular Dynamics and Binding Free Energy Calcs

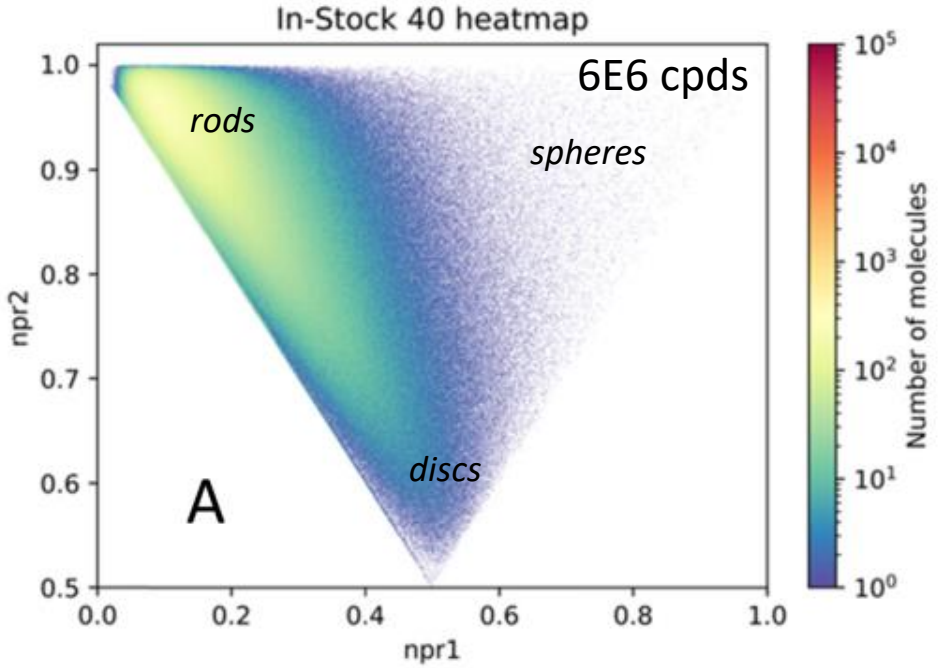
School of Medicine and Public Health
UNIVERSITY OF WISCONSIN–MADISON

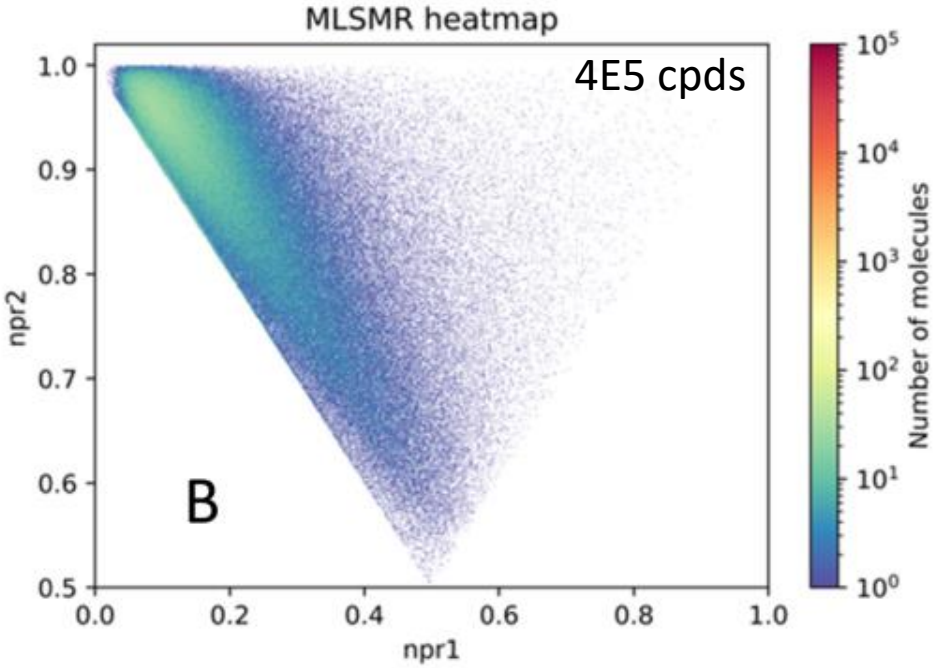
Contact me at SMSF

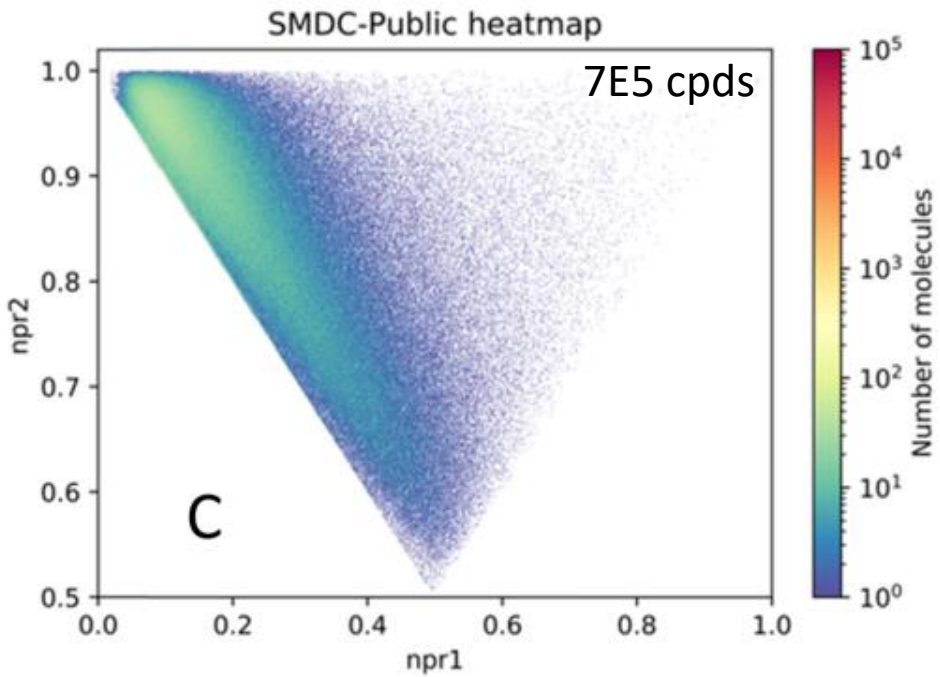
sserickson@wisc.edu

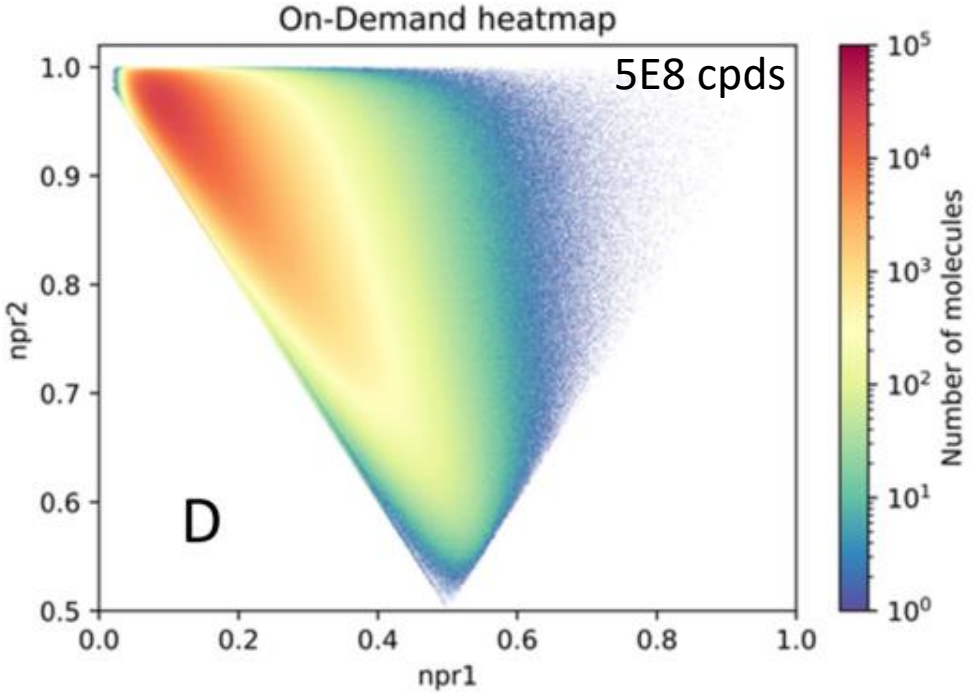
- Do you study a protein involved in a disease?
- Looking for molecules to modulate a protein's function?
- Need probes for studying your protein or mechanism?
- Do you study a biological process or phenotype you want to perturb?
- Do you have active compounds for your target but don't know what to do next?
- Would you like assistance interpreting HTS data and choosing the best active molecules to move forward?
- Want analogs of known active compounds for your target protein?
- Need help navigating the purchasable chemical space to acquire compounds for testing?

https://cancer.wisc.edu/research/resources/ddc/smsf/

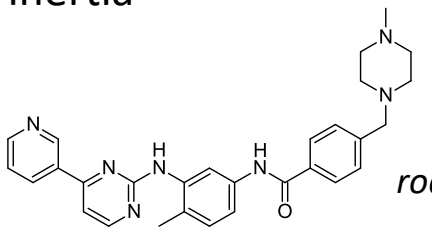




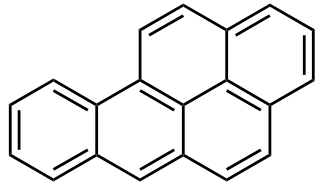




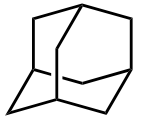
NPMI = normalized ratios of principle moments of inertia



rods



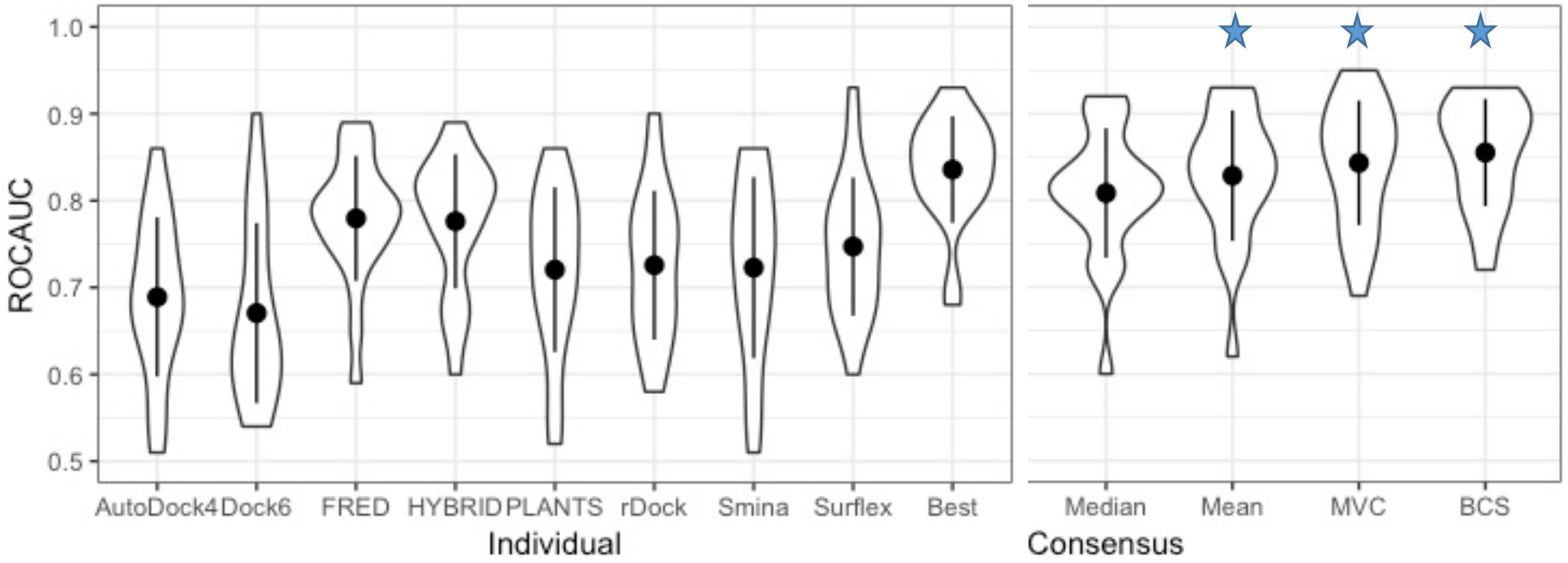
discs



spherical

Irwin JJ et al., "ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery." *J. Chem. Inf. Model*. 2020, **60**, 6065-6075.

Virtual screening performance on 21 benchmark targets



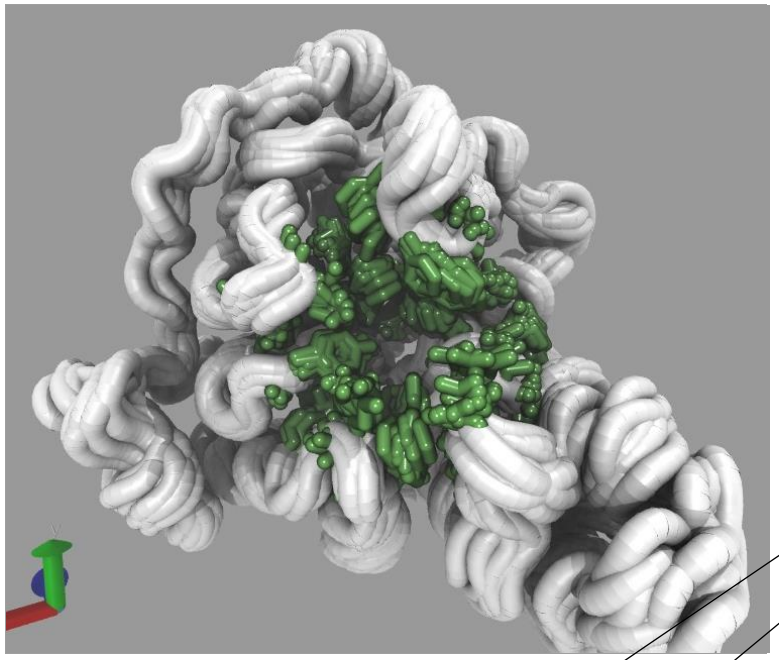
★ *P > 0.05*
Pairwise t-test

Target Class	Target
GPCR	ADRB1
GPCR	DRD3
Ion Channel	GRIA2
Kinase	BRAF
Kinase	CDK2
Kinase	PLK1
Kinase	SRC
Miscellaneous	FABP4
Receptor	ESR1
Receptor	ESR2
Other Enzymes	ACE
Other Enzymes	GLCM
Other Enzymes	HDAC8
Other Enzymes	HIVINT
Other Enzymes	PDE5A
Other Enzymes	PTN1
Protease	ADA17
Protease	FA10
Protease	HIVPR
Protease	MMP13
Protease	TRY1

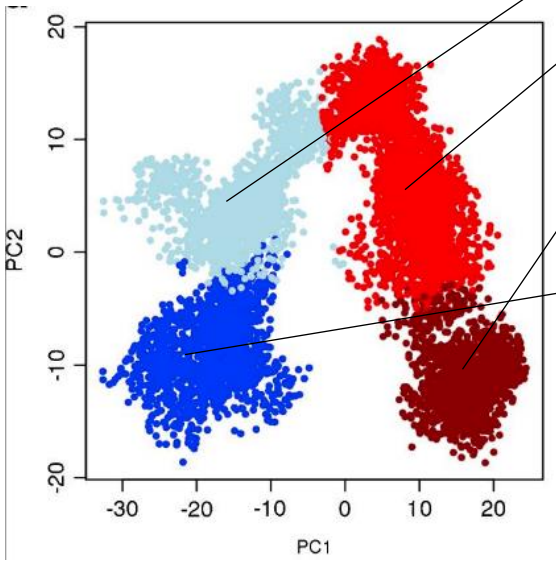
PI Mitchell (Gitter/Hoffmann co-Pis)
https://research.wisc.edu/funding/uw2020/round-3-projects/an-adaptive-computational-pipeline-drug-discovery/

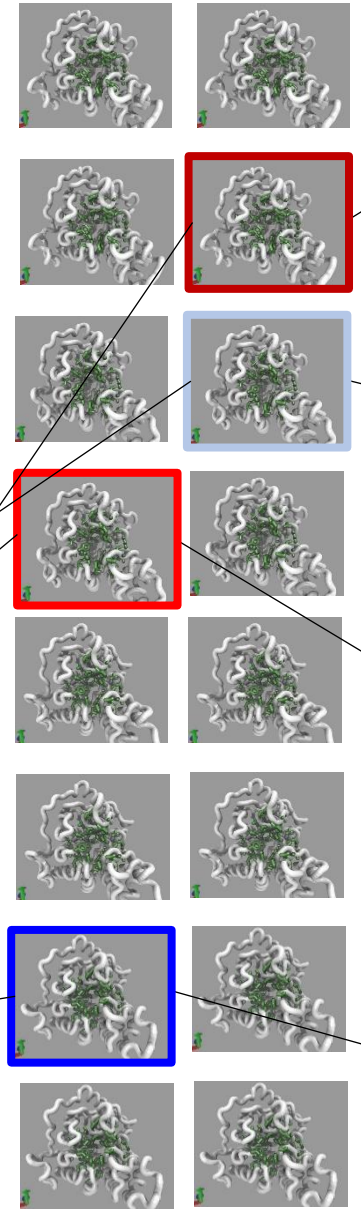
Ericksen et al., *J. Chem. Inf. Model*. **2017**, 57, 7, 1579-1590 DOI: 10.1021/acs.jcim.7b00153

Ensemble Docking

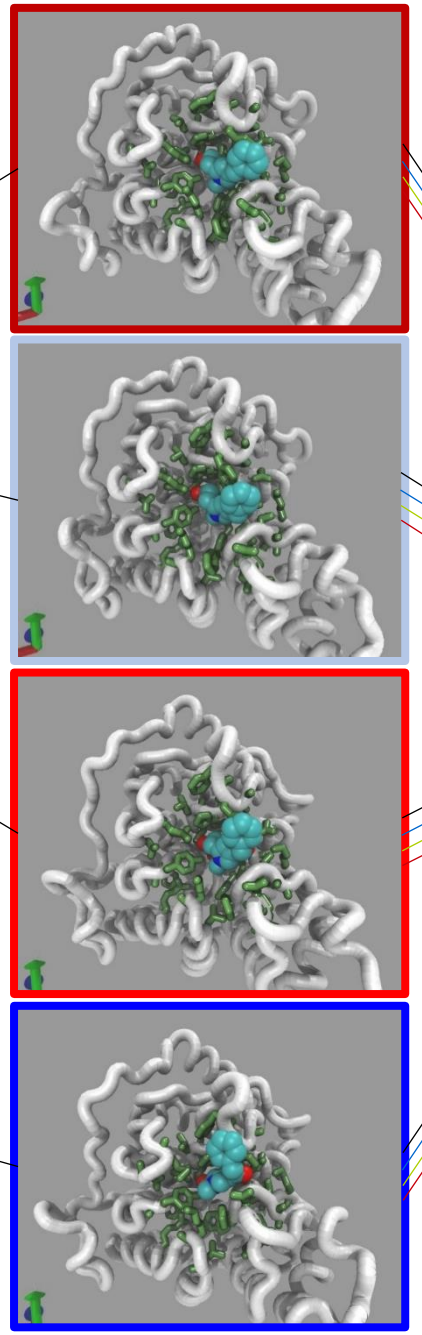


Enumerate





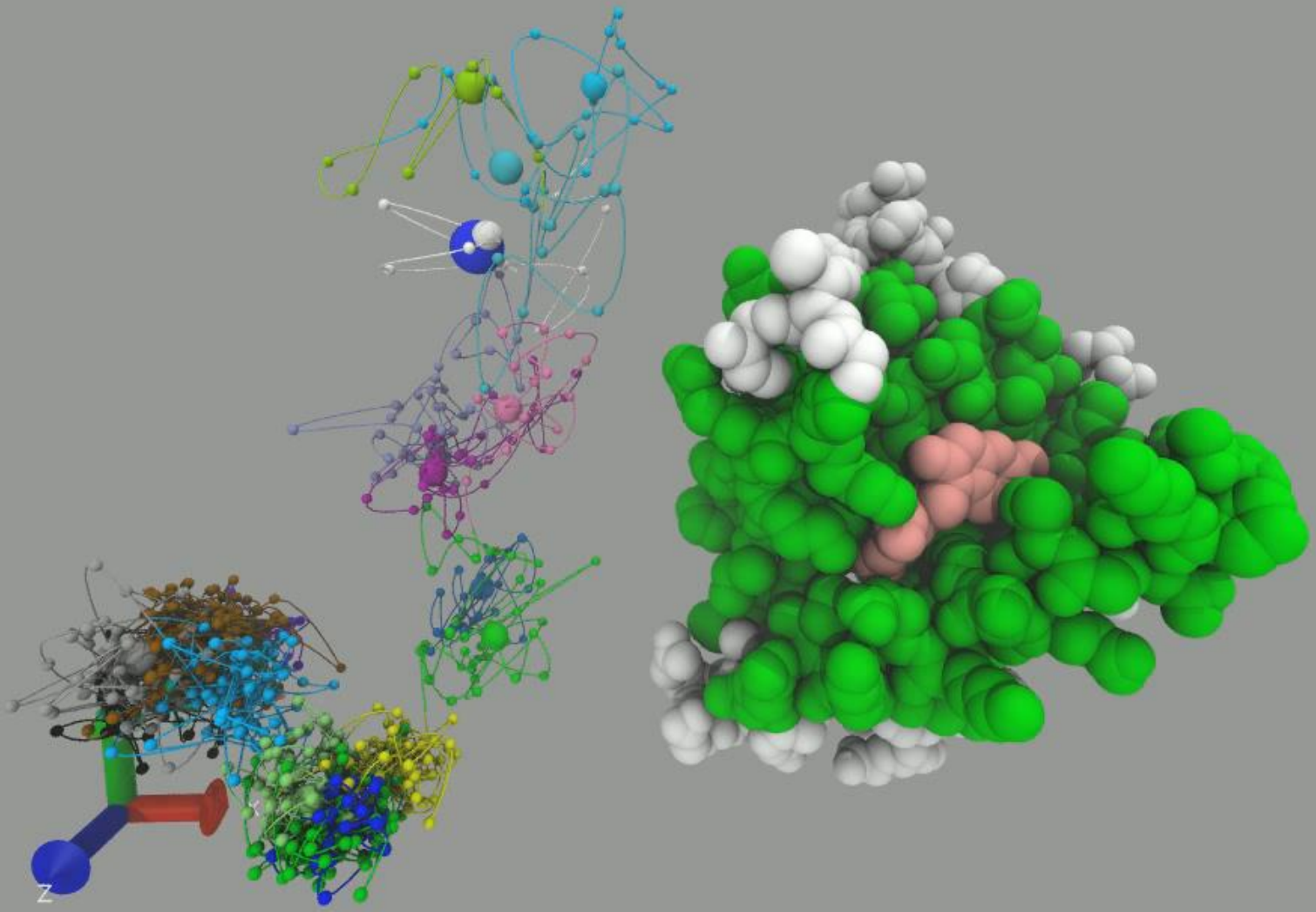
Select



Consensus Scoring

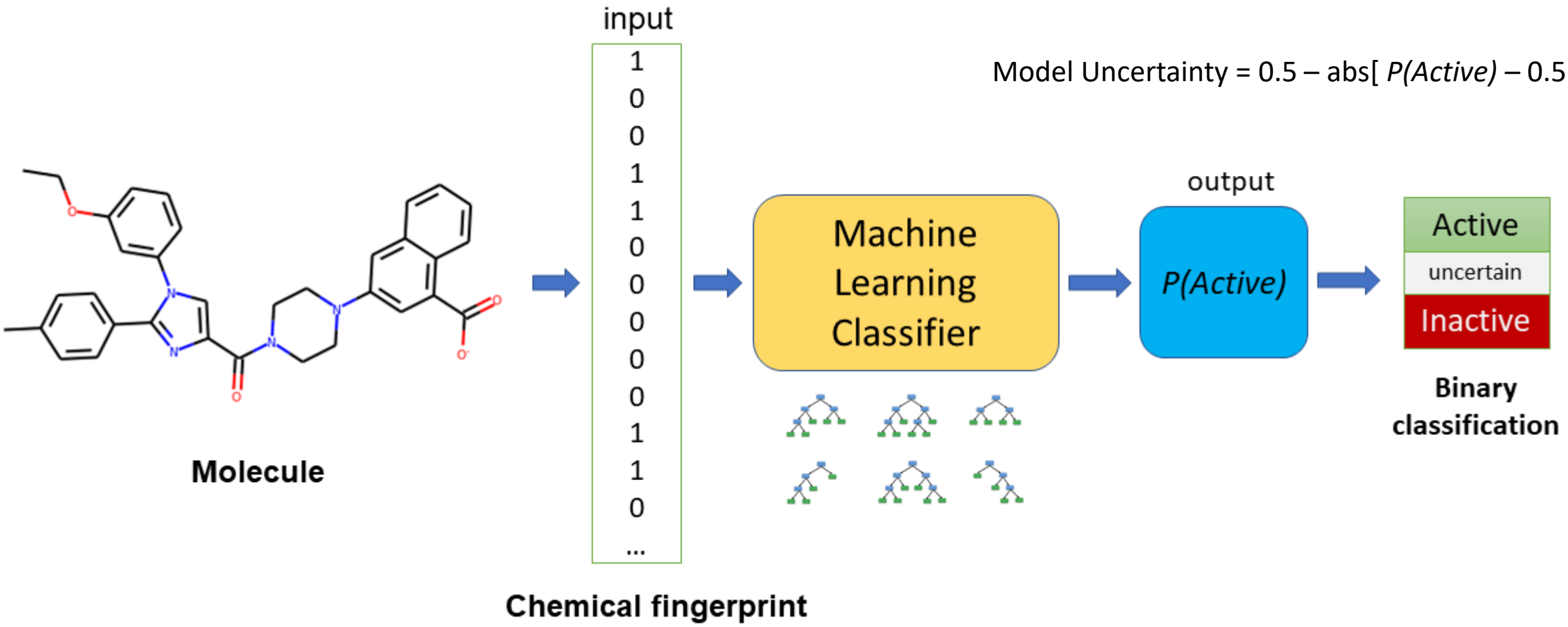
Dock with Multiple Programs, Extract Features

Ensemble Docking



hivint

Ligand-Based Virtual Screening—a ML hit-finding model



Model Uncertainty = 0.5 – abs[*P(Active)* – 0.5]

Gitter Lab: Liu, et al., "**Practical Model Selection for Prospective Virtual Screening.**" *J. Chem. Inf. Model.* 2019, 59, 1, 282−293. https://doi.org/10.1021/acs.jcim.8b00363

TIERED APPROACH FOR SBVS

Synthesized-on-demand virtual library
10 – 100 billion cpds (SMILES)

PROPERTY FILTER

1 – 10 billion cpds (SMILES)

ROCS

0.1 – 5 billion cpds (3D)

Fast Docking

5-200 million cpds

Consensus Docking

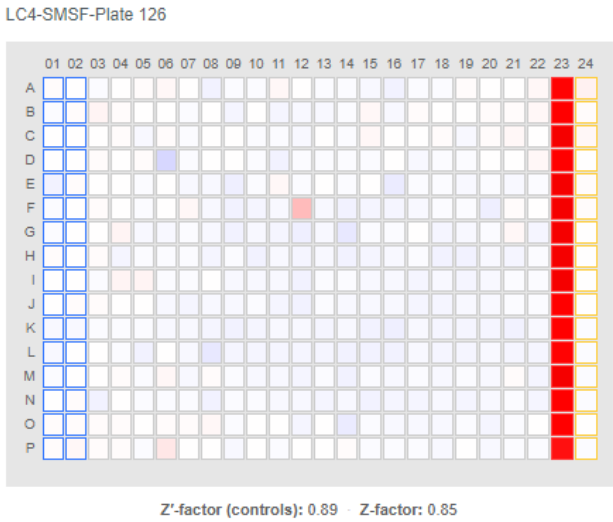
2-20k cpds

clustering

50-500 cpds

MD/ML-based affinity?

$$$



VS for ultra-large virtual libraries

- LBVS with RF and fingerprints easily scales to 1.0 billion cpds
- SBVS study required 5 million CPU hours to evaluate 1.3 billion cpds.
  - Gorgulla, C., et al., “An open-source drug discovery platform enables ultra-large virtual screens.” Nature 2020,580, 663–668.
- Another SBVS used 27,612 GPUs to score 1.37 billion in < 24 hours.
  - Glaser, J., et al., “High-throughput virtual laboratory for drug discovery using massive datasets.” The International Journal of High Performance Computing Applications.
- We have applied SBVS in consensus docking screens with 6 programs on libraries up to ~40 million cpds.

Abstract

Progress in chemical synthesis strategies has given rise to vast “make-on-demand” chemical libraries. Such libraries, now virtual, are bounded only by synthetic feasibility and grow exponentially. Making and testing significant portions of such libraries on a new drug target is not feasible. We increasingly rely on computational approaches called virtual screening methods to help us navigate large chemical spaces and to prioritize the most promising molecules for testing. The main challenge now is to scale existing virtual screening methods, or develop new ones, with sufficient molecule throughput and scoring accuracy to accommodate ultra-large compound libraries. Here I will describe some promising approaches that leverage high-throughput computing to meet this challenge.

What is *virtual screening*?

- **Drug Discovery**: looking for needle in haystack: potentially 10^{33} drug-like organic molecules* are theoretically possible—but only a some will be **active** against a disease state and also non-toxic
- **High Throughput Screening (HTS)**: Brute Force approach is to experimentally test 10,000s to millions molecules---expensive!
  - ~$15-200 compound (to purchase them)
  - ~$0.10-1.00 per experiment (to test them in automated, robotic HTS experiments)
- **Virtual Screening (VS)**: Filter very large molecule sets by computational techniques. Prioritize a subset for testing in a highly focused screen.

*Polishchuk PG, et al., JCAMD 2013 27(8):675-9

DUD-E: Benchmarking Data Set to Validate Docking-Based VS Methods

- "*A **D**atabase of **U**seful **D**ecoys: **E**nhanced*"
- 102 protein targets
- 22,886 active compounds with minimum potency 1 µM (or better)
- 100-600 ligands per target
- ~50 decoys for each active ligand (~2% actives)
- Decoys property-matched but dissimilar 2-D topology.
  - Properties: MW, LogP, HBA, HBD, rotatable bonds, net charge
  - ECFP4, keep 25% most dissimilar
- Actives are clustered. Diversity of actives is promoted by keeping max of 3 tightest binders in each cluster.

Mysinger et al., J. Med. Chem., 2012, Jul 5. doi 10.1021/jm300687e .

Docking programs have different search and scoring strategies

Docking Program	Search Algorithm	Scoring Function
AutoDock v4.2	Lamarkian Genetic Algorithm with Simulated Annealing	Forcefield
DOCK v6.7	Incremental Construction (Anchor-and-grow)	Forcefield
FRED v3.0.1	Exhaustive rigid docking search, discretized configuration space	Empirical
HYBRID v3.0.1	Exhaustive rigid docking search, discretized configuration space	Empirical + Knowledge-Based
PLANTS v1.2	Ant Colony Optimization	Empirical
rDock v2013.1	Genetic Algorithm, Monte Carlo, Minimization	Empirical
Smina (Vina) 1.1.2	Exhaustive flexible docking search, discretized configuration space	Knowledge-Based
Surflex v3.040	Incremental Construction with Matching Algorithm	Empirical

Docking

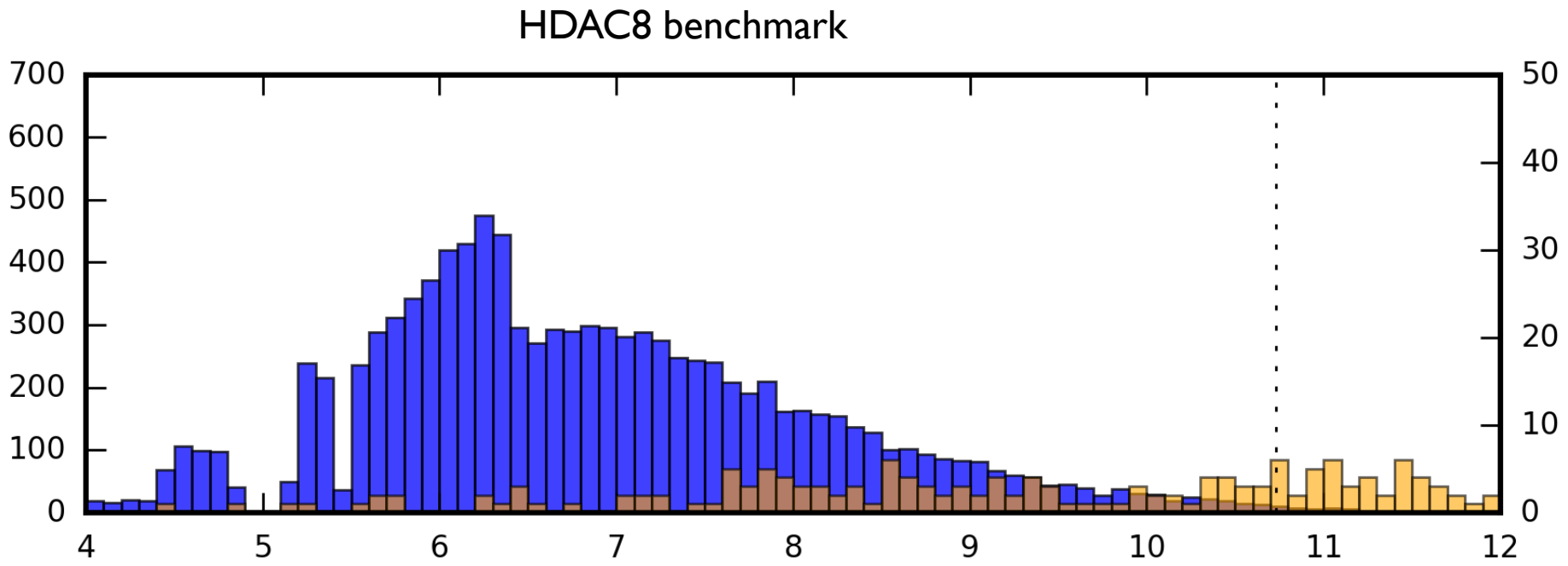
Scoring

No single program works for all targets

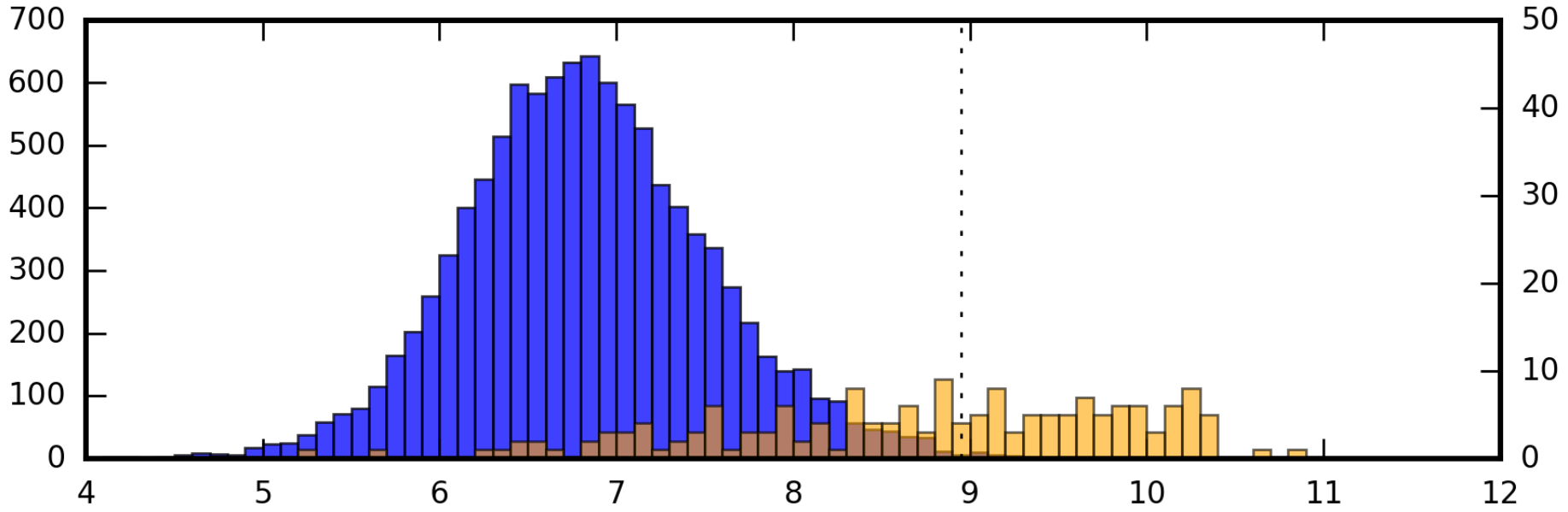
- No way to decide *a priori* which program is best for a new target

Individual Docking Algorithms

Target Class	Target	AD42	DOCK6	FRED	HYBRID	PLANTS	rDock	Smina	Surflex	Best
GPCR	ADRB1	0.68	0.78	0.77	0.65	0.86	0.81	0.79	0.80	0.86
GPCR	DRD3	0.69	0.59	0.79	0.81	0.69	0.66	0.68	0.71	0.81
Ion Channel	GRIA2	0.73	0.60	0.79	0.77	0.73	0.77	0.75	0.77	0.79
Kinase	BRAF	0.73	0.60	0.75	0.69	0.54	0.79	0.86	0.71	0.86
Kinase	CDK2	0.76	0.61	0.81	0.85	0.68	0.74	0.71	0.69	0.85
Kinase	PLK1	0.60	0.48	0.80	0.75	0.65	0.68	0.57	0.60	0.80
Kinase	SRC	0.65	0.64	0.65	0.66	0.52	0.68	0.67	0.66	0.68
Miscellaneous	FABP4	0.67	0.54	0.84	0.82	0.74	0.60	0.77	0.79	0.84
Receptor	ESR1	0.82	0.54	0.88	0.81	0.77	0.87	0.86	0.74	0.88
Receptor	ESR2	0.77	0.48	0.89	0.89	0.69	0.80	0.79	0.68	0.89
Other Enzymes	ACE	0.78	0.72	0.80	0.84	0.84	0.62	0.61	0.76	0.84
Other Enzymes	GLCM	0.55	0.60	0.70	0.81	0.64	0.77	0.51	0.79	0.81
Other Enzymes	HDAC8	0.70	0.90	0.87	0.76	0.82	0.71	0.86	0.83	0.90
Other Enzymes	HIVINT	0.54	0.65	0.74	0.60	0.76	0.67	0.81	0.66	0.81
Other Enzymes	PDE5A	0.68	0.65	0.84	0.82	0.79	0.78	0.74	0.66	0.84
Other Enzymes	PTN1	0.66	0.76	0.76	0.78	0.72	0.76	0.66	0.88	0.88
Protease	ADA17	0.51	0.40	0.59	0.69	0.58	0.58	0.54	0.70	0.70
Protease	FA10	0.86	0.81	0.79	0.82	0.80	0.90	0.84	0.76	0.90
Protease	HIVPR	0.63	0.66	0.74	0.78	0.79	0.64	0.74	0.81	0.81
Protease	MMP13	0.67	0.60	0.77	0.87	0.71	0.67	0.67	0.76	0.87
Protease	TRY1	0.79	0.82	0.80	0.83	0.81	0.74	0.75	0.93	0.93
	mean	0.69	0.64	0.78	0.78	0.72	0.73	0.72	0.75	0.84
	std. dev.	0.09	0.12	0.07	0.08	0.10	0.09	0.10	0.08	0.06



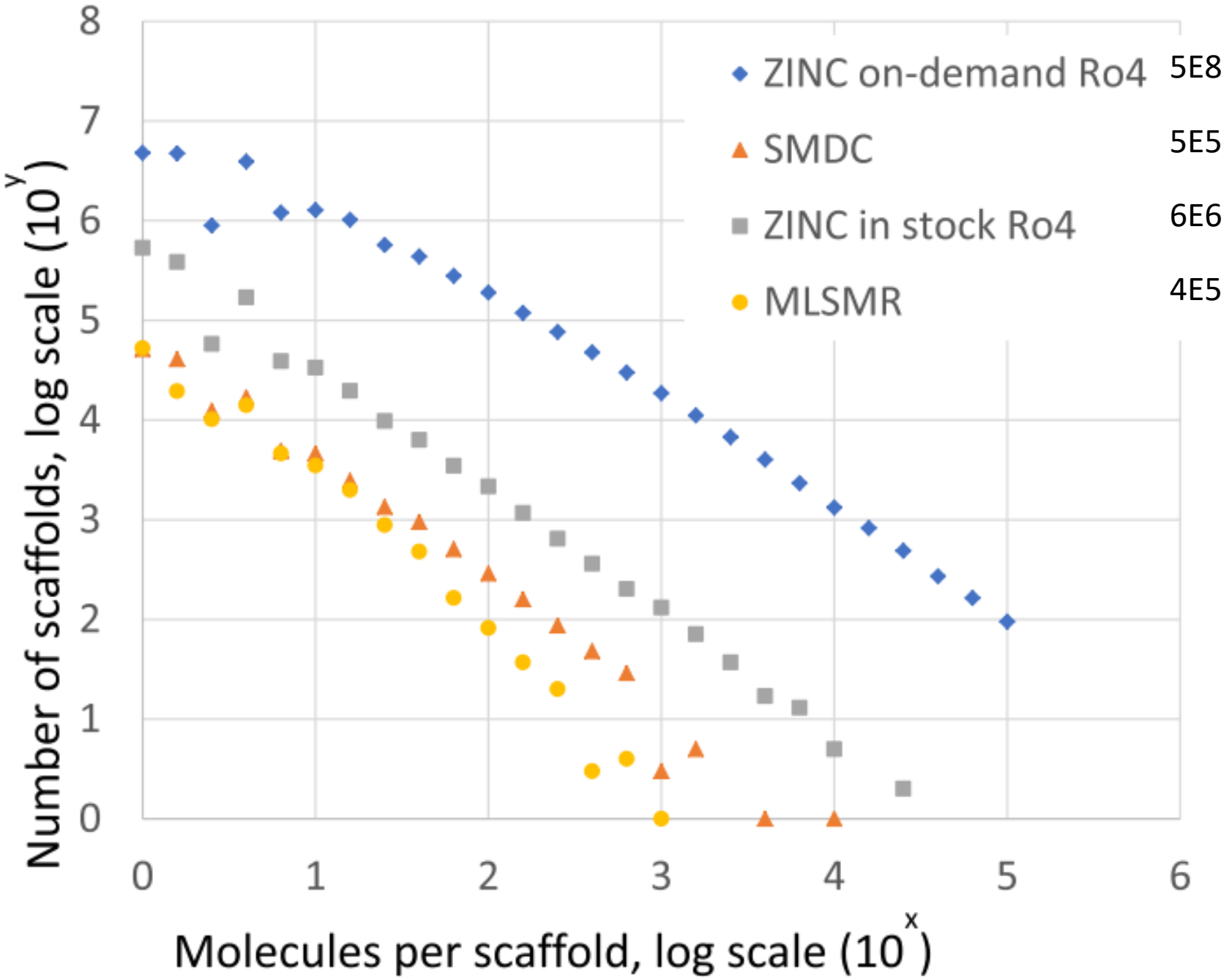
smina



consensus

Scaffold diversity

"in-stock" vs. "on-demand"



Irwin JJ et al., "ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery." *J. Chem. Inf. Model*. 2020, **60**, 6065-6075.

LBVS on Ultra-Large Virtual Chemical Library

- After scoring, request pricing info on top 10,000 ranked cpds
- Reduce list to 5620 cpds based on availability
- Clustered the 100,000 LifeChem, 1024 cpds from AMS, and 5620 Enamine REAL cpds.
- Use clusters to select a final list for purchase—predicted to be highly active, chemically diverse, chemically dissimilar from known actives in training set. Keep only Enamine clusters that did not contain an active cpds from training set or AMS set. (5620→2679 cpds)
- Run the following PAINS (RDKit), Inpharmatic, and Lipinski filters (2679→1604)
- Retain only cpds with >= 0.35 Tanimoto distance from closest training active (and AMS active) (1604 → 1354 cpds).
- Take best scoring rep from remaining clusters (1354→311 cpds)
- Select 100 diverse cpds from 311 using iterative Tanmioto distance selection. 1st selection as one with highest score. Second selection is the most distant from first. Continue selections of cpds most distant from the prior selections (mean or sum).
- Update quote for these 100, 90 are still available with pricing info. Other 10 required custom chemistry ($$$ and longer time)
- Ordered 90, synthesis failed on 22. Tested the 68 delivered in dose-response using 8 concentrations (in quadruplicate).
- 31 of 68 cpds tested were hits! (45.6% hit rate)

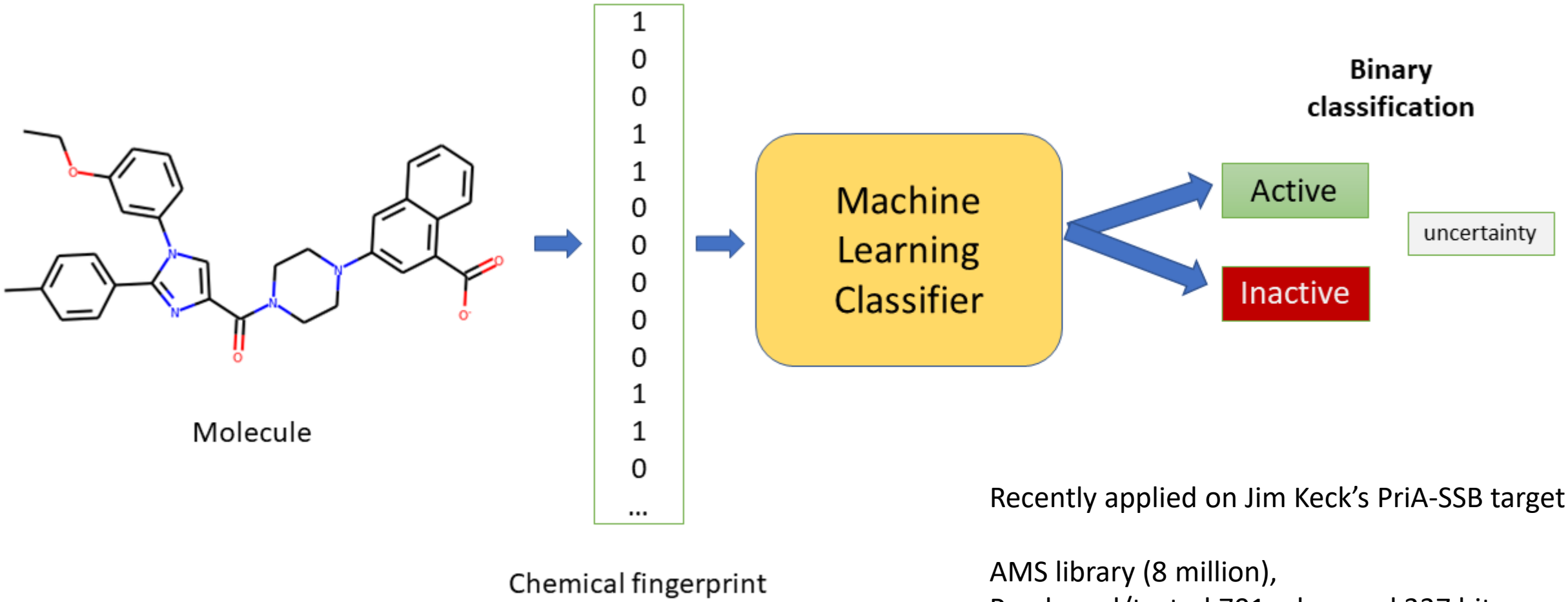
Gitter Lab: Alnammi M. et al., “Scalable supervised learning for synthesize-on-demand chemical libraries.” manuscript in prep

How do we scale LBVS to HTC resources?

- Each RF-based compound evaluation is independent--*pleasantly parallelizable*!
- Evaluation with pre-trained RF doesn't benefit from specialized hardware or multiple cores.

- To maximize throughput:
    - Split compound library up into chunks.
        - Each chunk should run in ~2hr (to efficiently use OSG/off-campus resources).
        - We split 1,077,562,987 cpds into 18 chunks of ~60 million (mean runtime=53.2 hrs, stdev=6.4 hrs)
    - Dock each chunk on a single slot to scavenge ANY open slots. Dock compounds in chunk serially.
    - Checkpointing is enabled and a wrapper script is used to track the compounds completed in case job is evicted and migrates to another node.

Ligand-Based VS—a ML hit-finding model



Recently applied on Jim Keck's PriA-SSB target:

AMS library (8 million),
Purchased/tested 701, observed 337 hits

Enamine REAL database (1.1 billion cpds)
Purchased/tested 68, observed 31 hits

Liu, et al., "**Practical Model Selection for Prospective Virtual Screening.**" *J. Chem. Inf. Model.* 2019, 59, 1, 282−293.
https://doi.org/10.1021/acs.jcim.8b00363