10th Users' Conference of IT4Innovations
5-6 November 2026, Ostrava

From Methods to Molecules: Four Years of Open-Source Computer-Aided Drug Design

Pavel Polishchuk

Group of chemoinformatics and drug design
Institute of Molecular and Translational Medicine
Faculty of Medicine and Dentistry
Palacky University
Czech Republic
pavlo.polishchuk@upol.cz
https://imtm.cz/laboratories/chemoinformatics-and-drug-design

The current state & our development principles

WHAT A MODERN CAMPAIGN MUST DO

Search chemical space at scale
Docking of millions to billions of structures

Refine docking with physics-based methods
Simulations of hundreds of protein-ligand complexes, to resolve what a rigid score cannot

Automation and reproducibility
Restartable, auditable, HPC-friendly runs

Missing: one open stack - design, dock, simulate - at scale, on different hardware, with no license

WHAT TODAY'S TOOLING DELIVERS

Licensing barriers
Per-seat and per-node fees

Fragmented workflows
Every stage is a separate tool and data format; ad-hoc glue scripts become the pipeline

Rebuilt, never reused
Each group re-implements similar infrastructure and pipelines for every new project

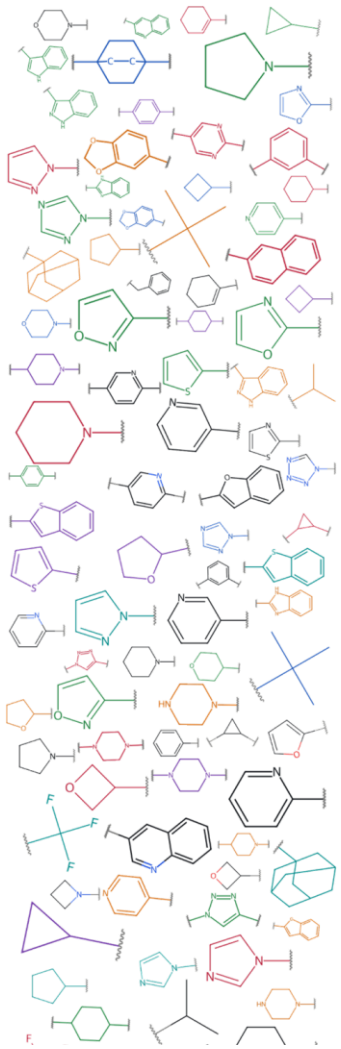
Lack of validation of components
Applicability and limitations of the incorporated modules and tools are not always established

OUR DEVELOPMENT PRINCIPLES

- **Open-source license** (mainly BSD-3, MIT)
- **Modular and extensible**
- **Easy to use and integrate** (CLI and API access)
- **Reproducible**
- **Scalable** (multi-core, multi-server, HPC clusters)
- **Resumable** (not wasting results on interrupted long runs)
- **Validated prospectively**
- **Continuous support** (we use tools internally)
- **Keep backward compatibility** (mainly for popular tools)

The virtual screening and de novo design pipelines

CReM fragment database (ChEMBL, Enamine, in-house)

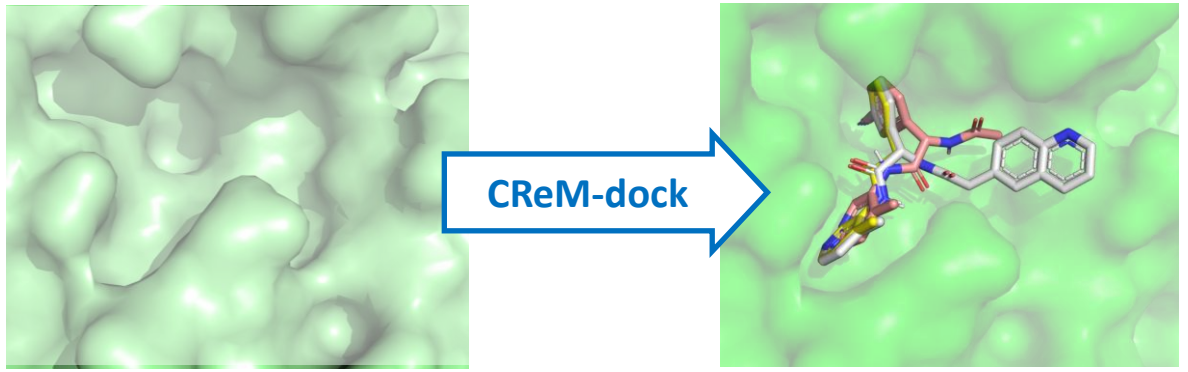


De novo design / optimization

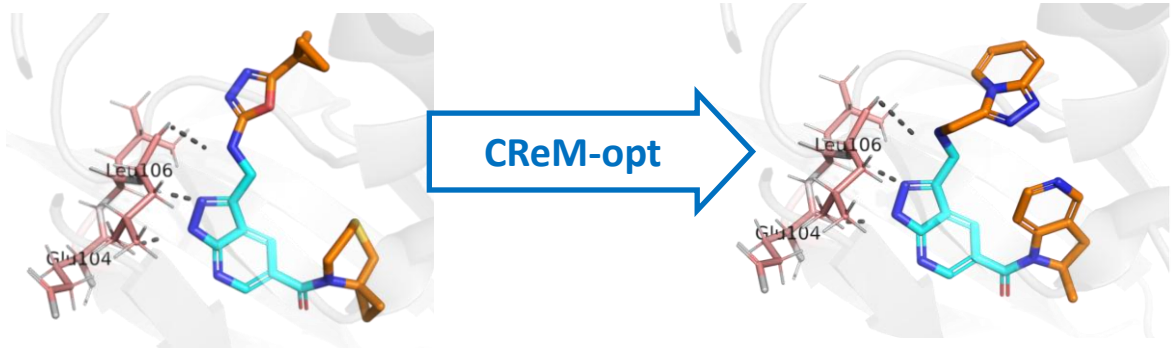
3D pharmacophore



binding site



structure for optimization



Virtual screening

Compound library 10^2-10^7 structures

Docking (**EasyDock**)

10^2-10^3 structures

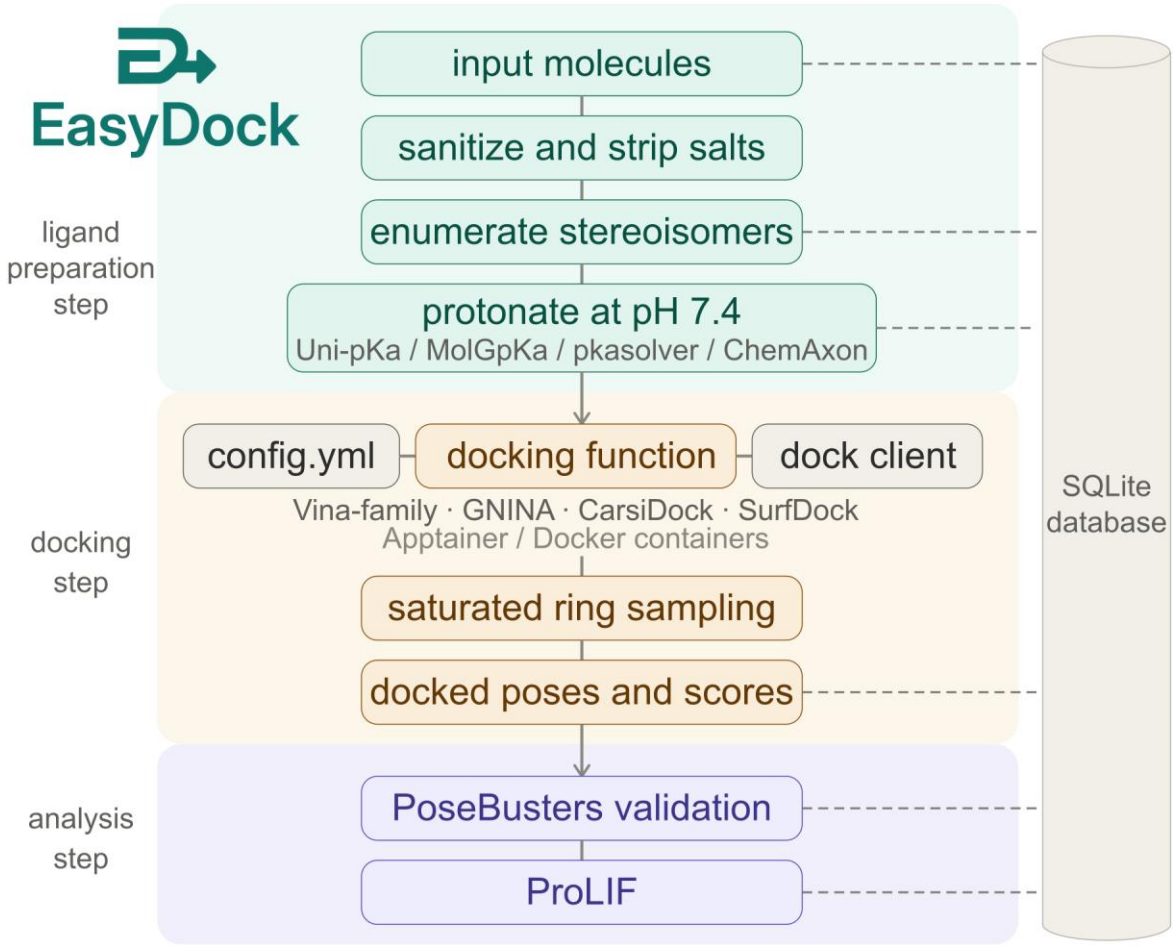
Rescoring: MM-GBSA (**StreaMD**)

10-500 structures

Rescoring: Absolute binding free energy (A3FE)

5-20 structures

EasyDock: scalable and customizable molecular docking



https://github.com/ci-lab-cz/easydock

Modular client-server architecture

simplify integration of new tools

One interface, many engines

AutoDock Vina, Smina, QVina2, GNINA, plus GPU Vina-family and deep-learning CarsiDock & SurfDock - orthogonal scoring for consensus ranking.

Built for scale

Prepare ligands once and reuse across targets/engines; distributed task handling; CLI and Python API for embedding in larger workflows.


Guzel Minibaeva


Veincent Yap


Dr. Aleksandra Ivanova

G. Minibaeva, A. Ivanova and P. Polishchuk, *J. Cheminf.,* **2023**, 15, 102.
G. Minibaeva, V. Yap and P. Polishchuk, *J. Chem. Inf. Model.,* **2026**, 66, 8691-8687.

EasyDock: scalable and customizable molecular docking

Command-line

```
easydock
  -i ligands.smi
  -o dock.db
  --protonation ~/imgs/unipka.sif
  --program vina
  --config config_vina.yml
  -c 4
  -v
```

config_vina.yml

```
protein: protein/5ufx_protein.pdbqt
protein_setup: protein/grid.txt
exhaustiveness: 16
seed: 0
n_poses: 5
ncpu: 4
```

Python API

```python
from easydock.run_dock import docking
from easydock.dock.vina_dock import mol_dock
from easydock.protonation import protonate_molgpka
from rdkit import Chem

# protonation
data = [('NCC(=O)O', 'glycine'), ('C1CNCCN1',  'piperazine'])

for smi, name in protonate_molgpka(data):
    mol = Chem.MolFromSmiles(smi)
    if mol:
        mol.SetProp('_Name', name)
        mols.append(mol)

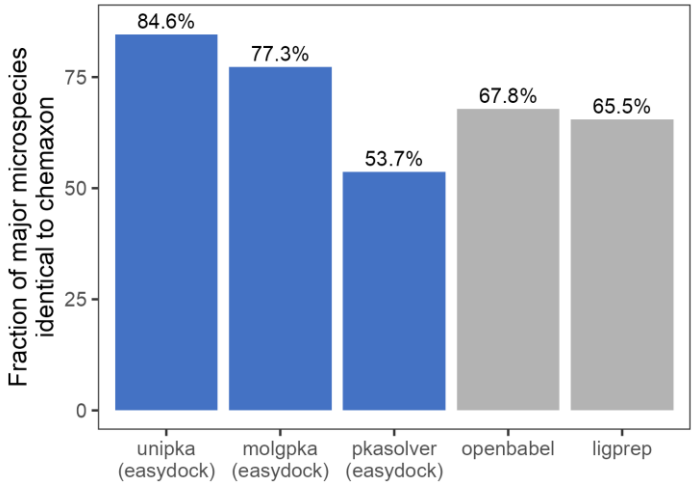
# docking
for mol_name, res in docking(mols, dock_func=mol_dock,
     dock_config='config.yml', ncpu=4):
     print(f'Molecule: 'mol_name')
     print(f'Results: 'res')
```

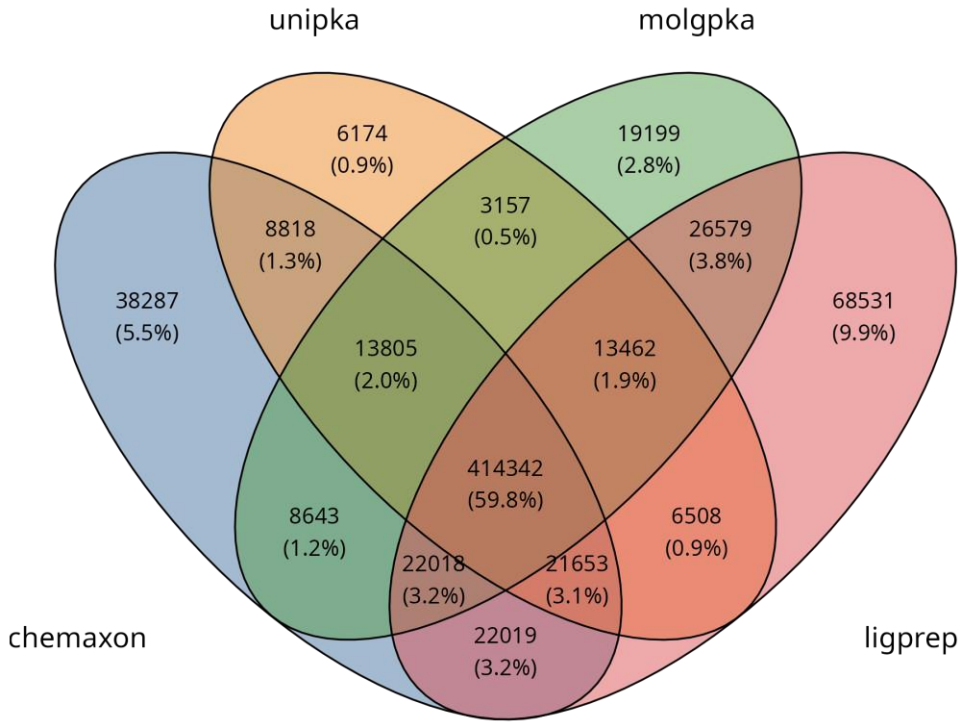
G. Minibaeva, A. Ivanova and P. Polishchuk, *J. Cheminf.,* **2023**, 15, 102.
G. Minibaeva, V. Yap and P. Polishchuk, *J. Chem. Inf. Model.,* **2026**, 66, 8691-8687.

EasyDock: scalable and customizable molecular docking

Program	Protonation wall time of 10000 molecules (CPU utilization)
OpenBabel	2 s (1 CPU)
MolGpKa	782 s (CPU node) (72%) 306 s (GPU node) (75%)
pkasolver	171 s (CPU node) (48%)
Uni-pKa	1781 s (CPU node) (93%) 834 s (GPU node) (77%)

Protonation of 807,407 structures from ChEMBL33





Docking run times for a set of 819 ligands of estrogen receptor 1 on CPU (128 cores) and GPU nodes (16 cores, 1 A100). Vina, QVina and Gnina used exhaustiveness 16.

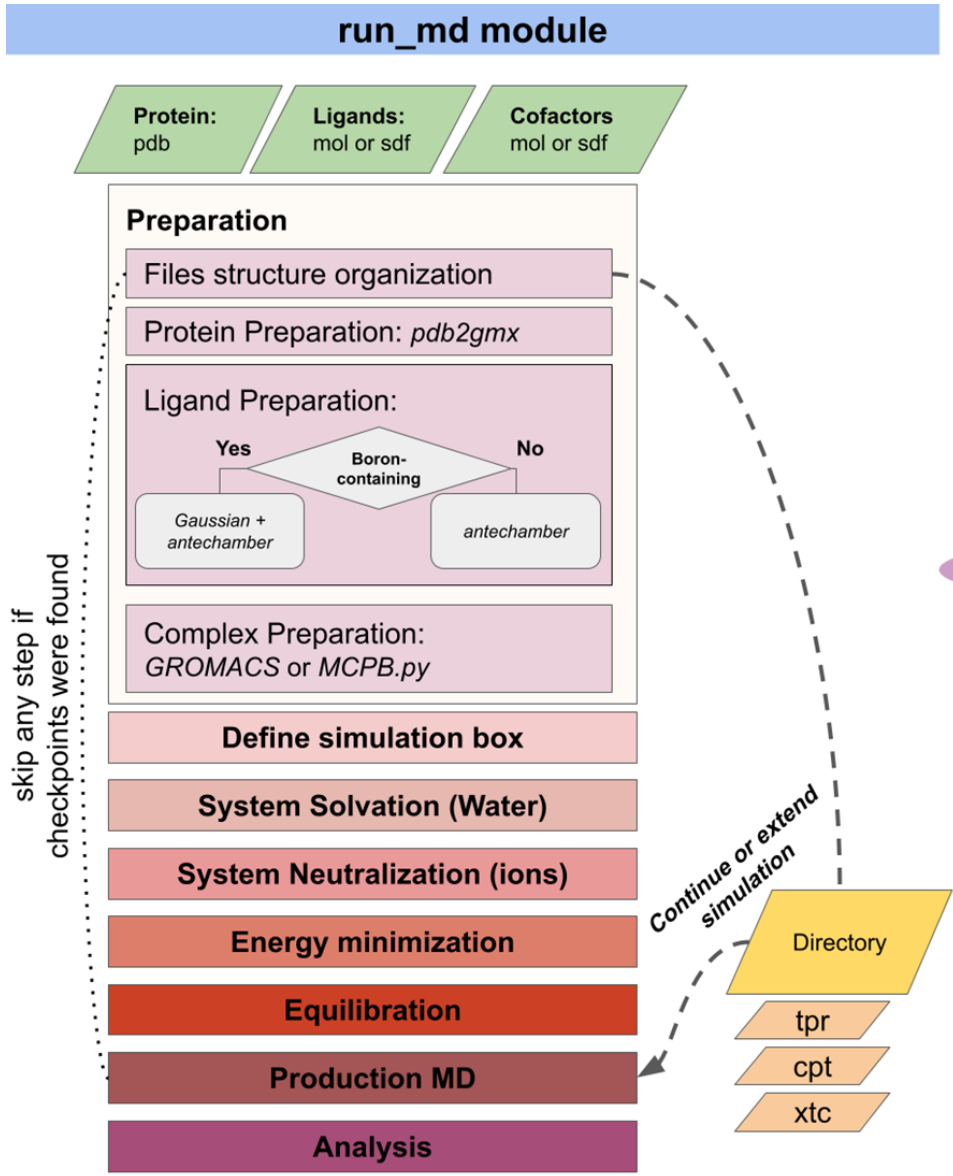
Program	Compute node	Docking wall time (h:mm:ss)	CPU utilization	Median docking time of a single ligand	PoseBusters
Vina	CPU	0:12:30	66%	19.1 s (4 cores)	99.4 %
QVina	CPU	0:05:48	85%	13.9 s (4 cores)	99.4 %
Vina-GPU	GPU	0:31:13	4%	1.7 s	99.0 %
GNINA	GPU	0:41:50	88%	32.9 s	95.4 %
CarsiDock	GPU	0:51:59	65%	3.8 s	41.1 %
SurfDock	GPU	2:15:14	75%	10.1 s	17.8 %

StreaMD: classical MD simulations at scale

Default force field (AMBER99SB-ILDN) can be changed by the user.

Position restraints are automatically applied to the heavy atoms of all protein, ligand and cofactors.

By default**, system Minimization** proceeds until the maximum force value reaches **1000.0 kJ/mol/nm** or less, but not exceeding 50000 steps. Following this, consecutive **1000 ps NVT** and **NPT equilibrations** are executed (the time duration can be customized by a user).





Automatic analysis

System preparation analysis:
potential energy, temperature, pressure, density
Trajectory analysis:
RMSD (protein, active site, cofactors and ligands), RMSF, radius of gyration

MMPBSA/GBSA

Prepare arguments and calculate MMP(G)BSA energy using **gmx_MMPBSA** tool

ProLIF

Get protein-ligand interactions from MD trajectories using **ProLIF** module

https://github.com/ci-lab-cz/streamd

A. Ivanova, O. Mokshyna and P. Polishchuk, *J. Cheminf.*, **2024**, 16, 123.

StreaMD: classical MD simulations at scale



Dr. Aleksandra Ivanova

To install:

pip install streamd

To run simulation:

run_md -p protein.pdb -l ligands.sdf --md_time 10

To extend simulation:

run_md --wdir_to_continue md_files/md_run/protein_ligand_*/ --md_time 20

G(P)BSA calculations:

run_gbsa --wdir_to_run md_files/md_run/protein_ligand_*

ProLIF calculations:

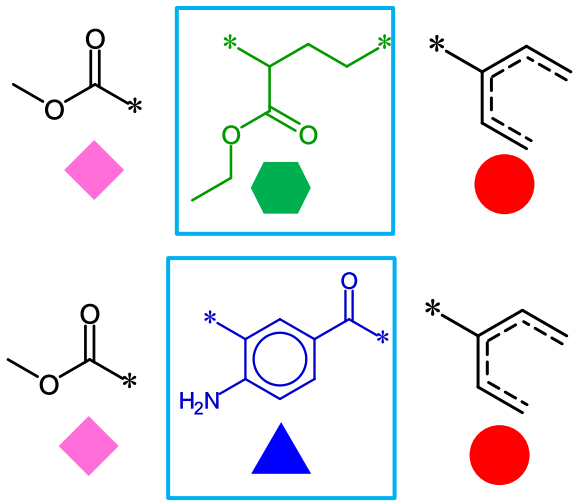
run_prolif --wdir_to_run md_files/md_run/protein_ligand_*

```
md_files/
├─ md_preparation/
│   ├─ protein/
│   ├─ ligands/
│   └─ cofactors/
└─ md_run/
    ├─ protein-id_ligand-id
    ├─ …
    …
```

```
run_md -h
usage: run_md [-h] [-p FILENAME] [-d WDIR] [-l FILENAME] [--cofactor FILENAME] [--clean_previous_md] [--hostfile FILENAME]
              [-c INTEGER] [--mdrun_per_node INTEGER] [--device cpu] [--gpu_ids GPU ID [GPU ID ...]] [--ntmpi_per_gpu int]
              [--topol topol.top] [--topol_itp topol_chainA.itp topol_chainB.itp [topol_chainA.itp topol_chainB.itp ...]]
              [--posre posre.itp [posre.itp ...]] [--protein_forcefield amber99sb-ildn] [--noignh] [--md_time ns]
              [--npt_time ps] [--nvt_time ps] [--seed int] [--no_dr] [--not_clean_backup_files] [--steps [STEPS ...]]
              [--mdp_dir Path to a directory with specific mdp files] [--wdir_to_continue DIRNAME [DIRNAME ...]]
              [-o OUT_SUFFIX] [--deffnm preffix for md files] [--tpr FILENAME] [--cpt FILENAME] [--xtc FILENAME]
              [--ligand_list_file all_ligand_resid.txt] [--ligand_id UNL] [--activate_gaussian module load Gaussian/09-d01]
              [--gaussian_exe g09 or /apps/all/Gaussian/09-d01/g09/g09] [--gaussian_basis B3LYP/6-31G*]
              [--gaussian_memory 120GB] [--metal_resnames [MN ...]] [--metal_cutoff 2.8] [--metal_charges {MN:2, ZN:2, CA:2}]
```

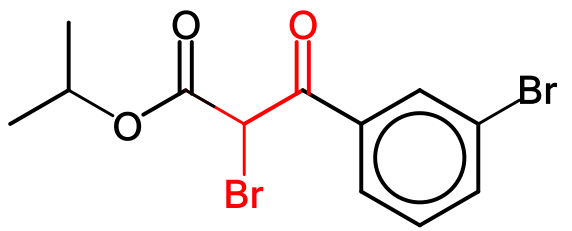

A. Ivanova, O. Mokshyna and P. Polishchuk, *J. Cheminf.*, **2024**, 16, 123.

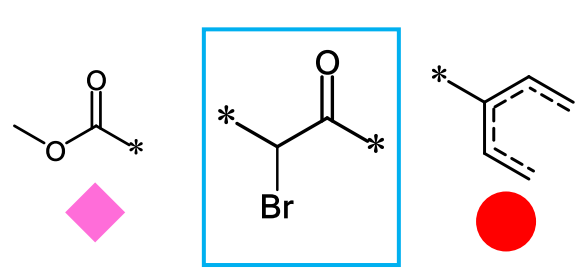
CReM: chemically reasonable mutations



CReM DB
(interchangeable fragments)

environment (radius = 3)	fragments
◆ ●	⬢ ▲ ...
...	...





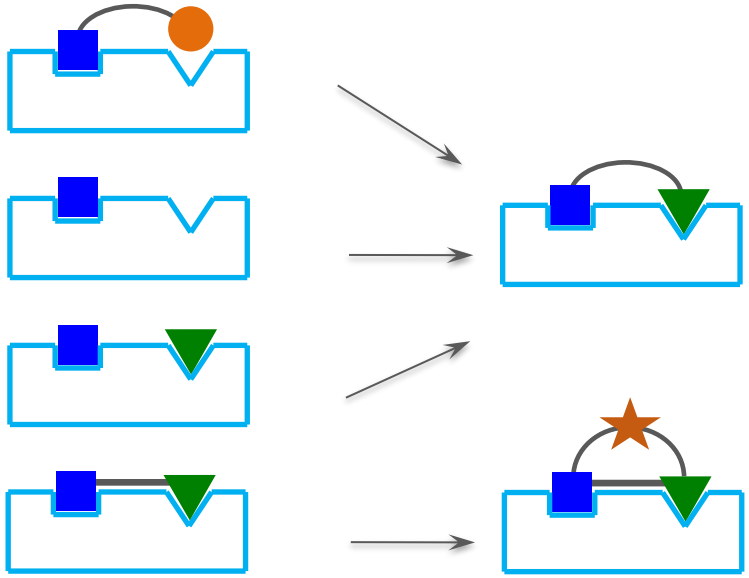
Modes

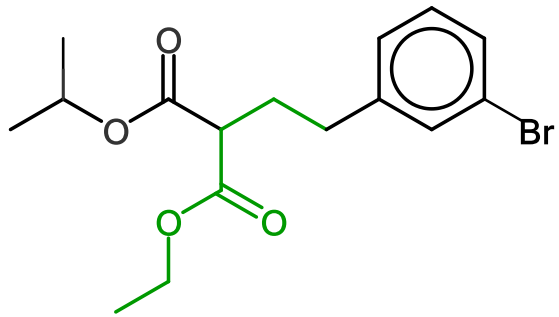
MUTATE

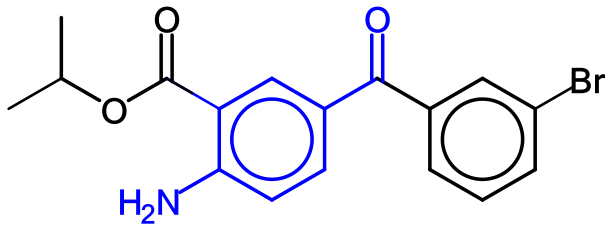
GROW

LINK

CYCLE







https://github.com/DrrDom/crem

P. Polishchuk, *J. Cheminf.*, **2020**, 12, 28.

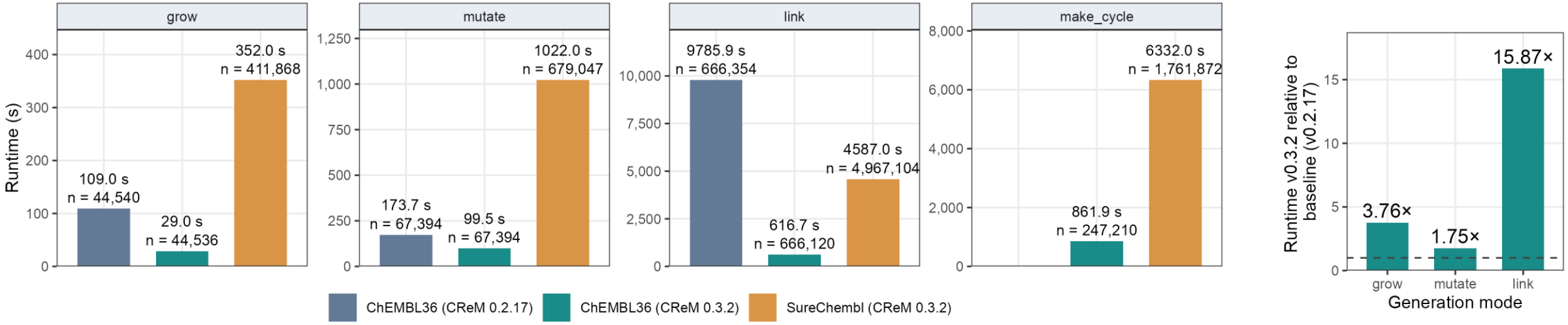
CReM: chemically reasonable mutations

Fragment database generation

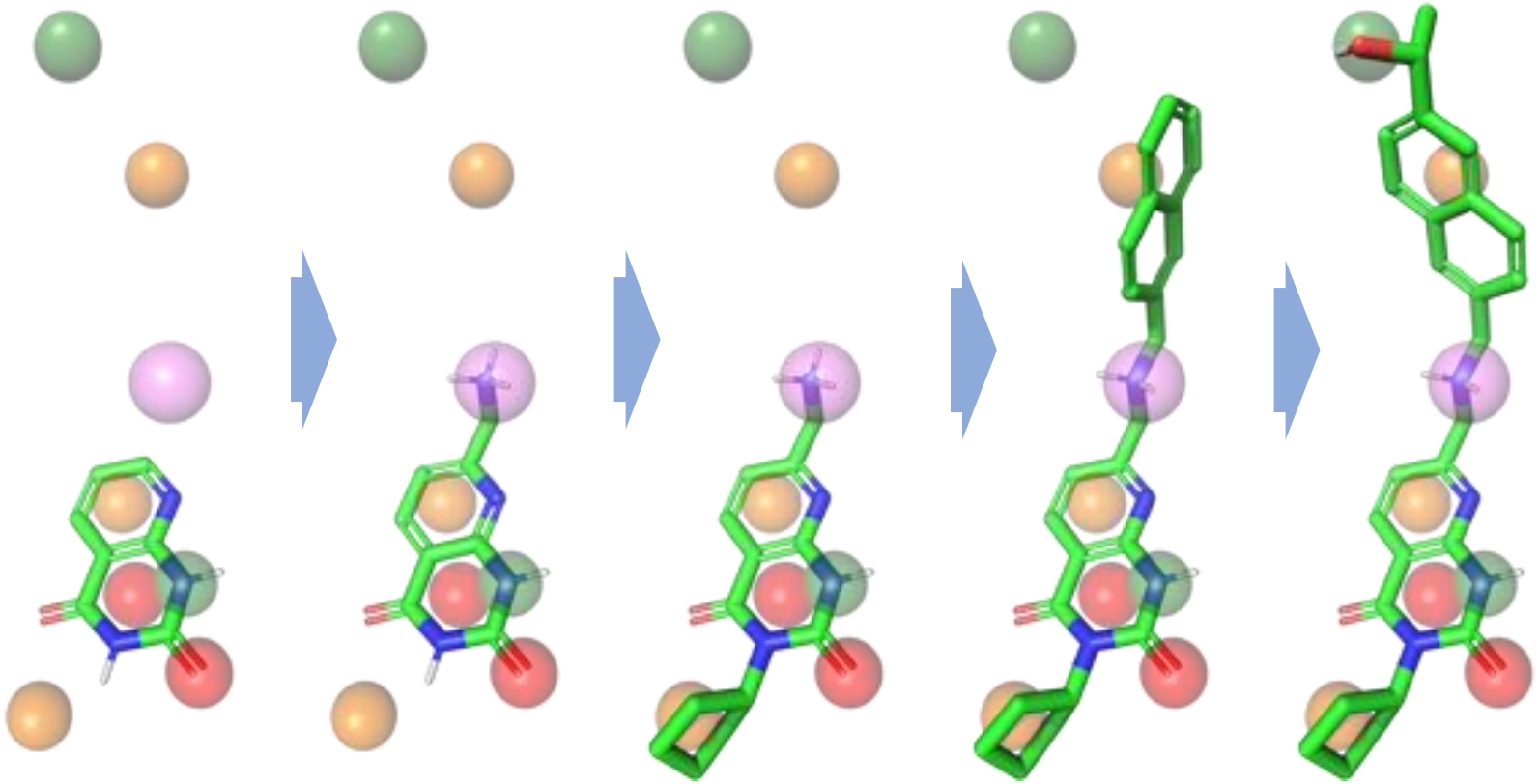
Source of fragments	CReM version	Fragmentation mode (--frag-mode)	SQLite database size	Wall time generation	Average CPU utilization
ChEMBL36 (2.72 M structures)	0.2.17	acyclic	61 GB	18.6 h	9 %
	0.3.2	acyclic	19 GB	3.8 h	50 %
	0.3.2 (12 parallel shards with 600 threads requested in total)	acyclic	19 GB	3 h	67 %
		acyclic + ring	28 GB	5.1 h	50 %
SureChembl (26.5 M structures)		acyclic + ring	321 GB	70.6 h	45 %

In-house node: 24 x AMD EPYC 9474F 48-core CPUs and 512 GB RAM; 350 cores were allocated for each database-generation run

Structure enumeration (single core mode)



CReM-pharm: de novo growing guided by 3D pharmacophore models





Alina Denzler



Dr. Dinesh Kumar

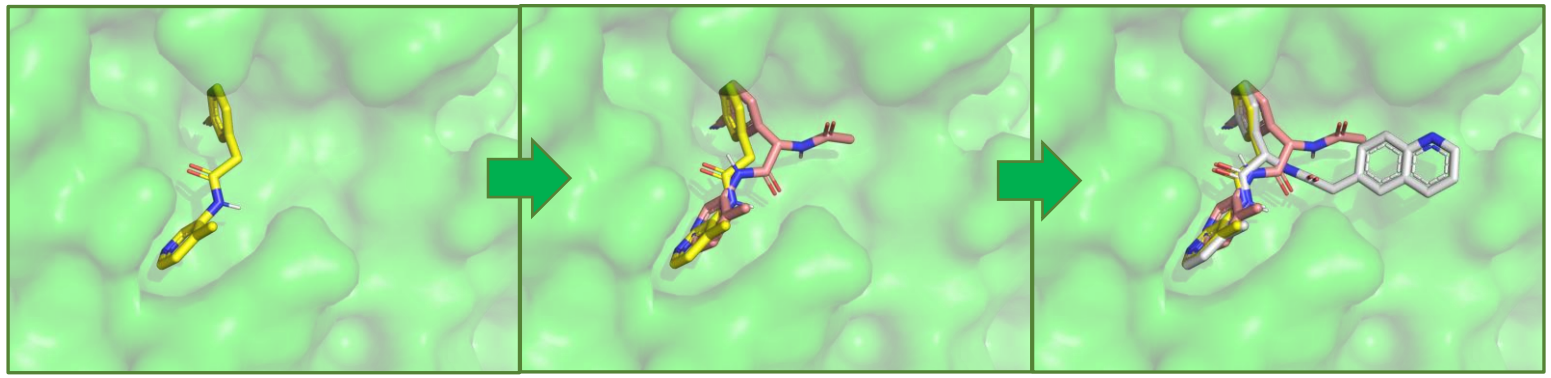
https://github.com/ci-lab-cz/crem-pharm

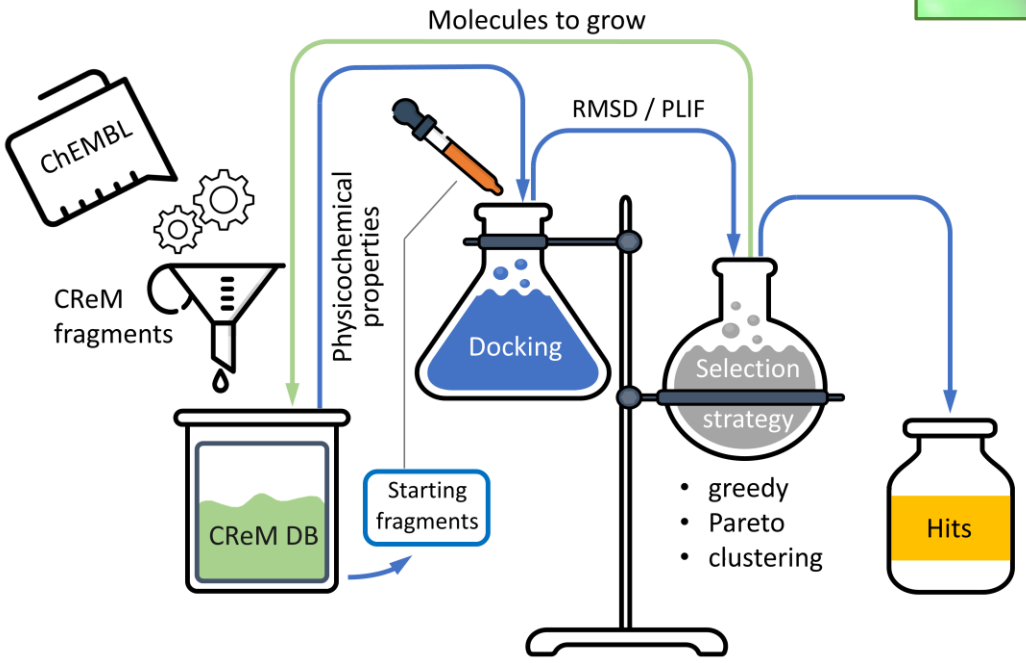
A. Denzler, D. K. Sriramulu, J. Pecha and P. Polishchuk, *J. Cheminf.,* **2026**, 18, 70.

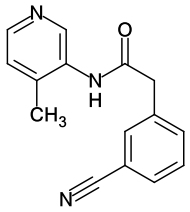
CReM-dock: de novo growing guided by molecular docking



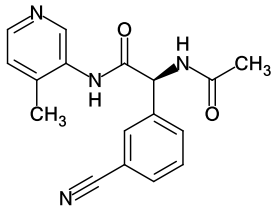
Guzel Minibaeva



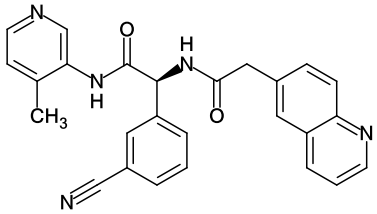




-6.2



-7.0



-9.4

https://github.com/ci-lab-cz/crem-dock

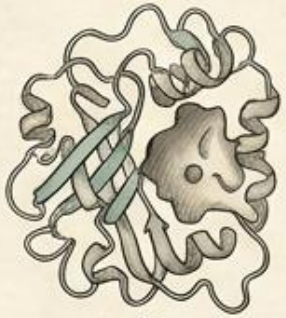
G. Minibaeva, H. Du, F. Clark, J. Michel and P. Polishchuk, *Digital Discovery,* **2026**, 5, 2271-2291.

Case study 1:
searching for CDK16 inhibitors in ultra-large databases

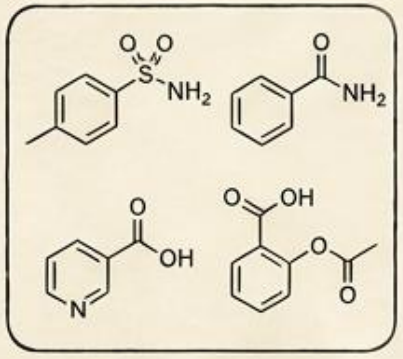
Design the Bait. Fish the Ocean.

1. De novo generation

Create seed molecules



Target protein



Seeds (bait)

2. Similarity search in ultra-large libraries



Find similar molecules (the catch)

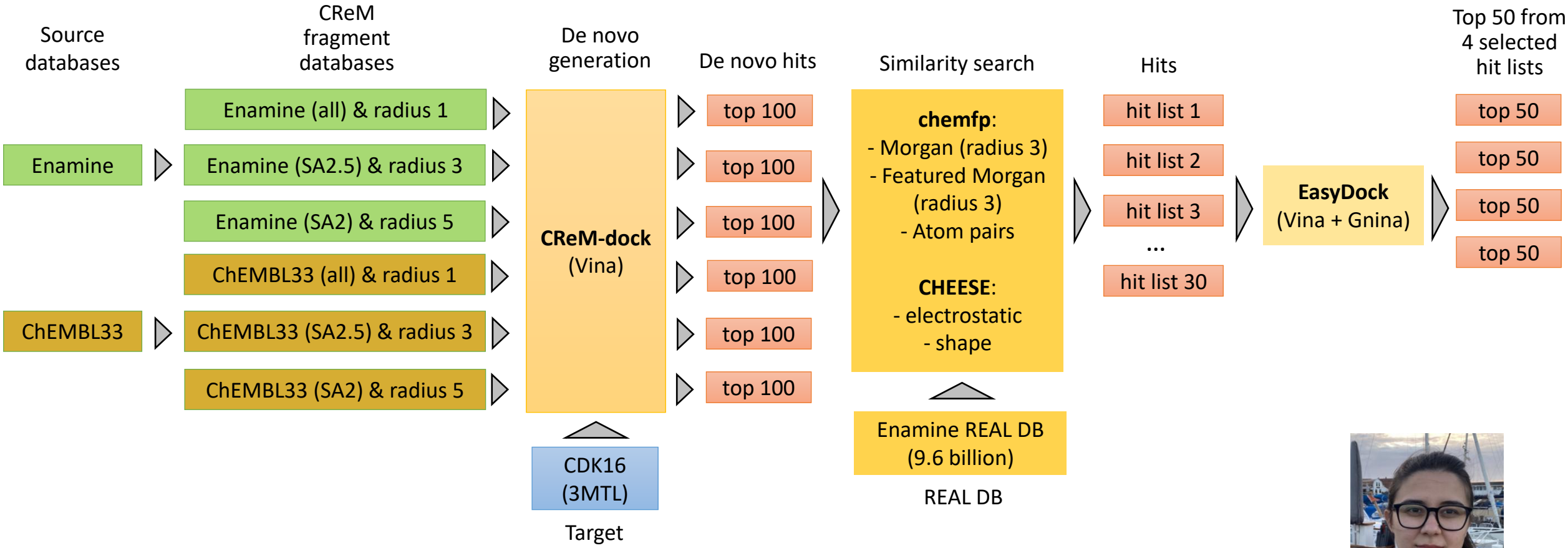
3. Prioritize & test

Select best hits for experiments



CReM-VS: searching in ultra-large databases guided by de novo designed hits

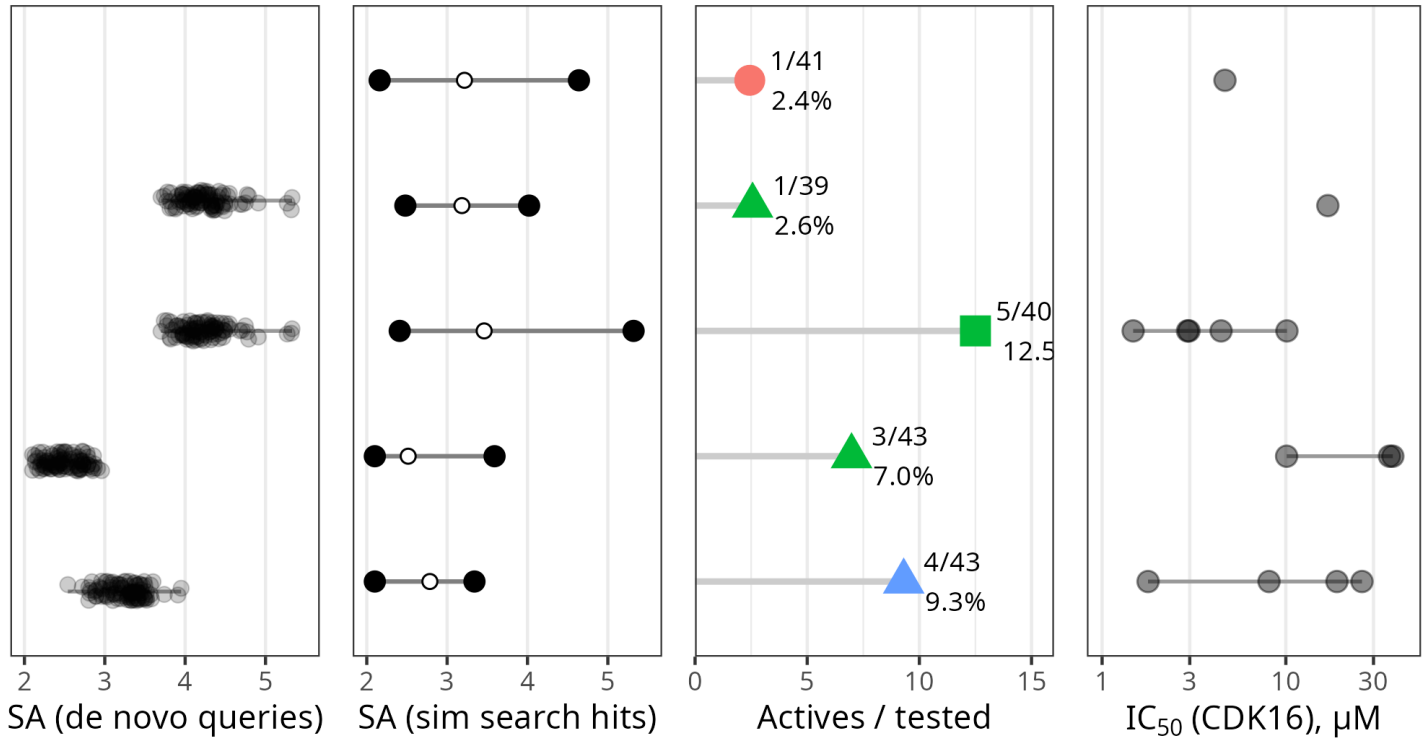
CDK16 study



Guzel Minibaeva

CReM-VS: searching in ultra-large databases guided by de novo designed hits

	Fragments	SA filter	Radius	Fingerprint
Virtual screening	—	—	—	—
CReM-VS	ChEMBL33	all	1	2D Atom pairs
CReM-VS	ChEMBL33	all	1	3D Electrostatic
CReM-VS	ChEMBL33	SA ≤ 2	5	2D Atom pairs
CReM-VS	Enamine	SA ≤ 2.5	3	2D Atom pairs



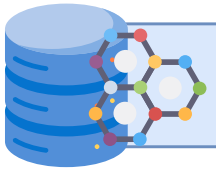
1. Poor synthetic accessibility of de novo queries does not affect downstream similarity search
2. 3D electrostatic fingerprints were better than 2D structural fingerprints
3. Source of fragment may have importance on activity of found hits

Case study 2:
discover novel DYRK1b inhibitors

De novo design of DYRK1b inhibitors



Stanislav Tricolici



CReM fragment DB (Enamine, radius 5)

Pharmacophore pre selection & HAC ≤ 12·& predicted docking ≤ −5 & Chiral centers ≤ 1

29 250 starting fragments

CReM-pharm
Pharmacophore-seeded growing

EasyDock

top 1000

CReM-dock
Docking-guided growing

CReM-opt
Genetic-algorithm optimization

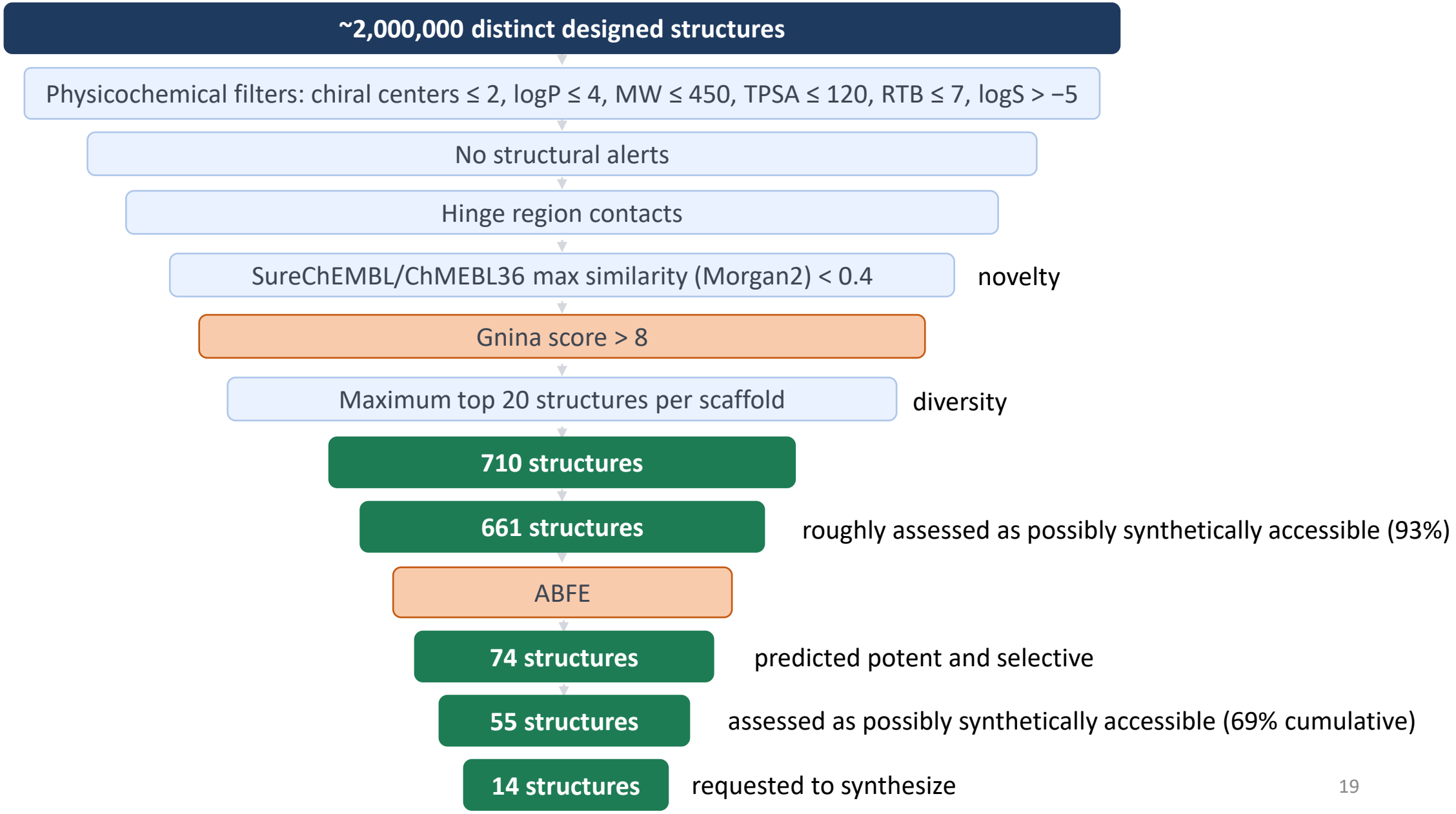
Generated Molecules

Physicochemical & structural filters + docking score

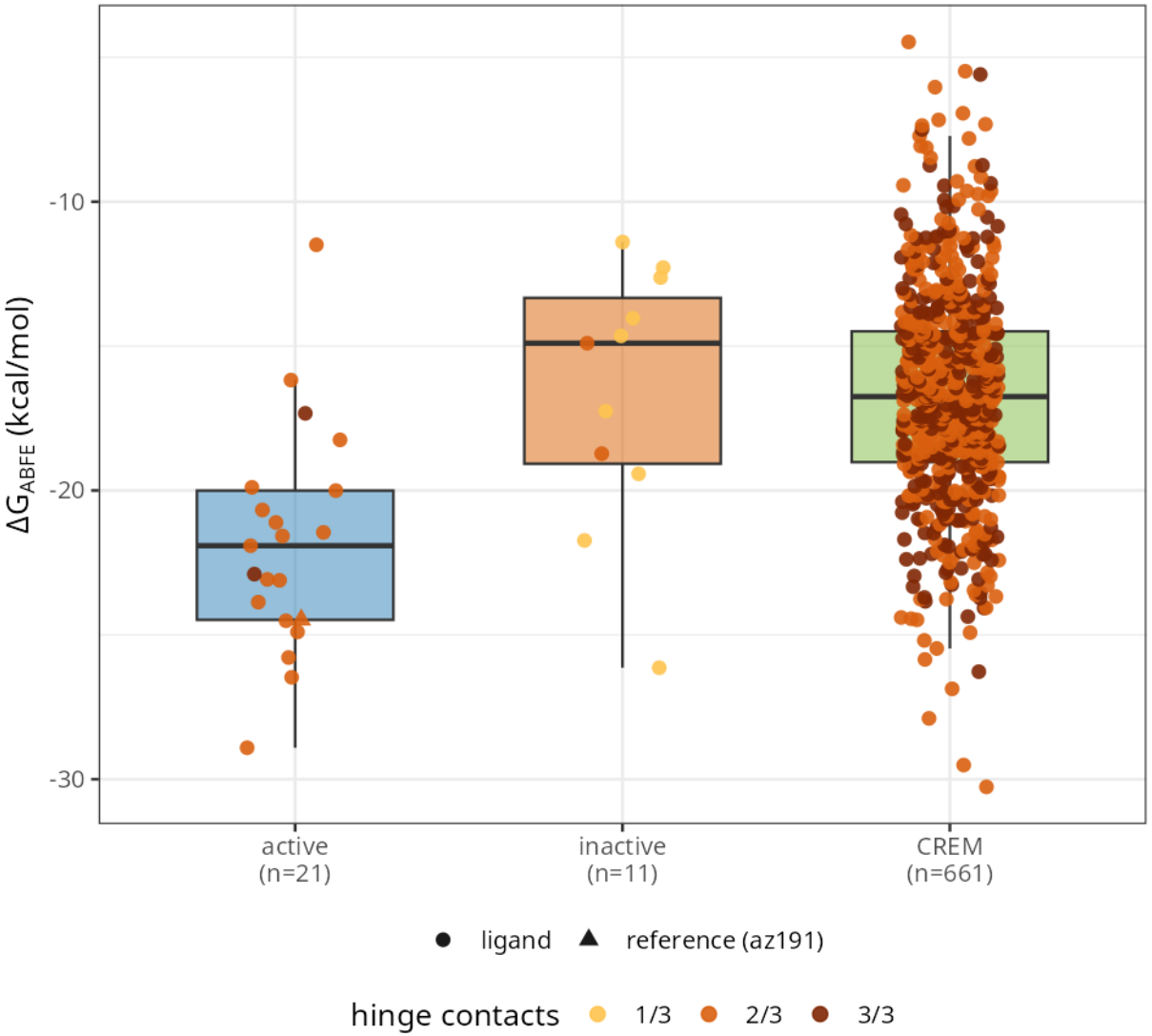
A3FE
(Absolute binding free energy)

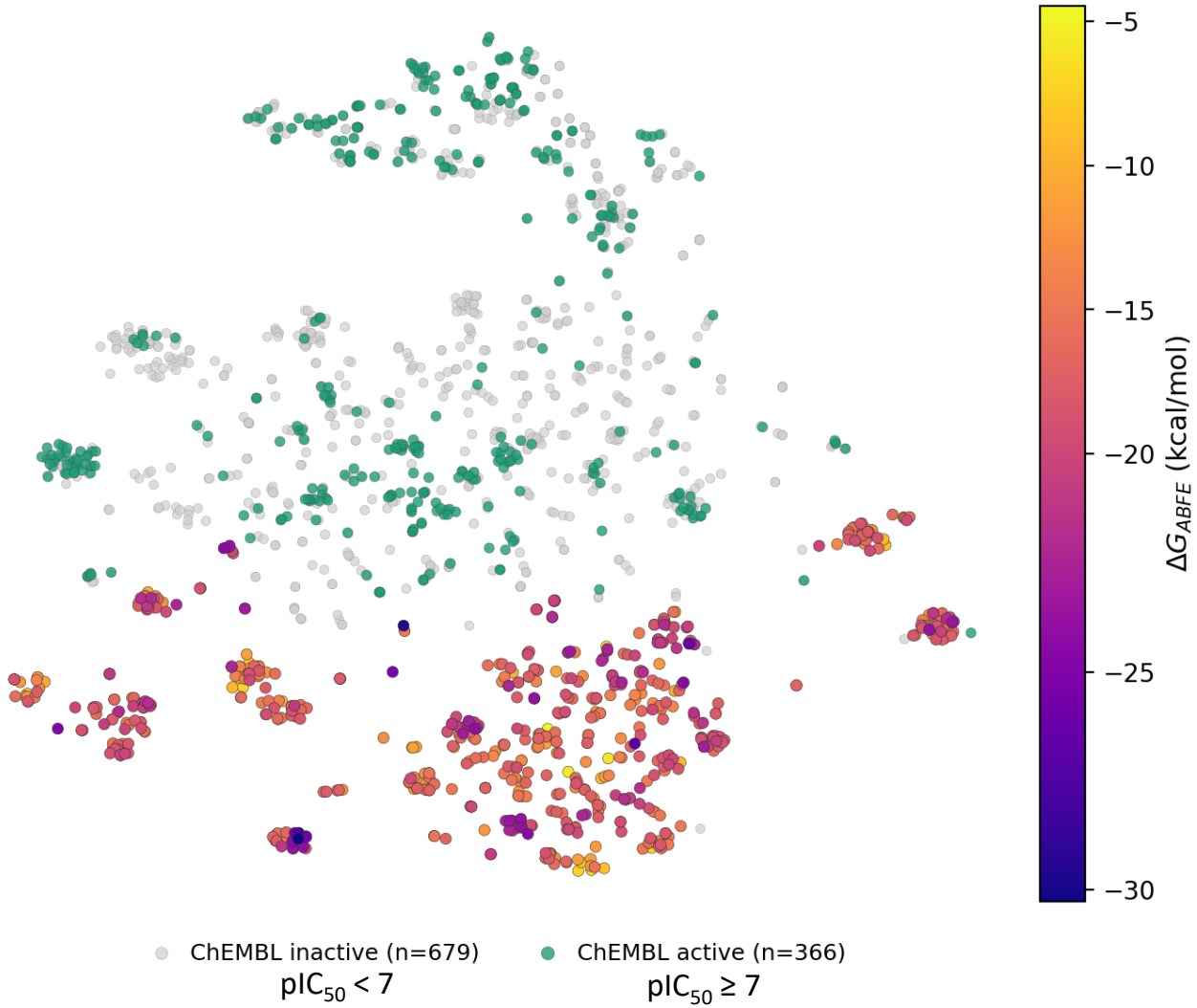
Synthesis of promising candidates

De novo design of DYRK1b inhibitors



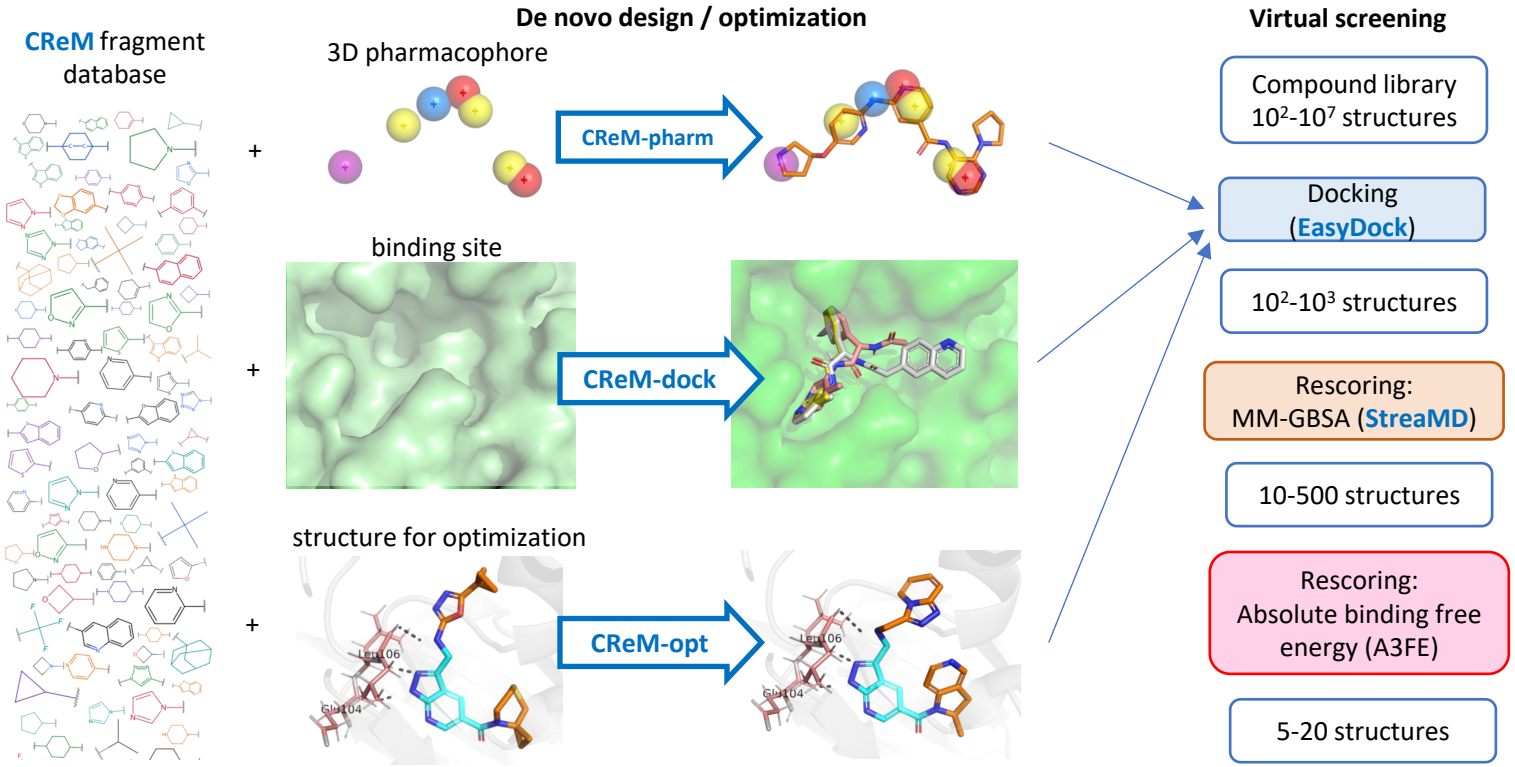
De novo design of DYRK1b inhibitors





Conclusions

- The tools were developed to be modular, extensible, computationally efficient, and easy to use through both CLI and API interfaces. Tools are comprehensively documented and designed to be accessible to students and researchers with limited prior experience.
- The tools are interoperable, can be combined into higher-level workflows, and they use compatible database schemas to simplify downstream analysis.
- In both prospective case studies, compound selection was fully automated and based exclusively on predefined computational criteria, allowing evaluation of the pipelines independently of medicinal-chemistry intuition or manual intervention.



Conclusions

Virtual screening

EasyDock – automated scalable molecular docking

Features:

- automated ligand preparation & docking
- different docking engines
- extensible client-server architecture
- post-docking analysis

Molecular dynamics

StreaMD – automated scalable classical molecular dynamics simulations

Features:

- GROMACS engine
- fully automated workflow
- post-MD analysis
- MM-PBSA/GBSA calculation

De novo design / optimization

- **CReM** – fragment based structure generation
- **CReM-pharm** – de novo design guided by 3D pharmacophore models
- **CReM-dock** – de novo design and scaffold decoration guided by molecular docking
- **CReM-opt** – de novo design and optimization guided by molecular docking

Features:

- exploration of ultra-large chemical libraries and beyond
- high structural novelty
- tunable synthetic accessibility

IMTM

Dr. Aleksandra Ivanova
Guzel Minibaeva
Stansilav Tricolici
Alina Denzler
Dr. Dinesh Sriramulu
Dr. Petr Dzubak
Dr. Martina Kintlova
Dr. Sona Gurska
Doc. Marian Hajduch

External

Veincent Yap
Haolin Du
Dr. Finlay Clark
Dr. Julien Michel

Dr. Andrew Dalke

DeepMedChem
Inte:ligand

https://github.com/DrrDom/crem
https://github.com/ci-lab-cz/easydock
https://github.com/ci-lab-cz/streamd
https://github.com/ci-lab-cz/crem-pharm
https://github.com/ci-lab-cz/crem-dock
https://github.com/ci-lab-cz/crem-opt

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the INTER-EXCELLENCE and INTER-EXCELLENCE II programs (LUARF18013 and LUAUS23262) and through e-INFRA CZ (ID:90254) and EHPC-BEN-2025B12-028.