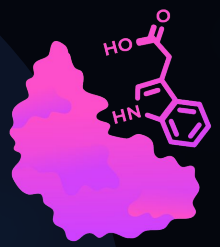
VantAI



plinder

Protein-Ligand INteraction Dataset and Evaluation Resource

Vladas Oleinikovas | RDKit UGM | 2024, Zurich

gs://plinder github.com/plinder-org plinder.sh

Outline

Motivation: why do we need PLINDER?

What is PLINDER and how it is created?

- Ingest
- Compare
- Split
- (Train)
- Evaluate

What is next?

- Further planned improvements
- MLSB/PLINDER Challenge
- Training Workshop

PLINDER is a result of a cross-collaboration effort!

It is ongoing but we welcome new contributors, too!

Why we need “another” Protein-Ligand dataset?

Many tasks that use ML models* trained on protein-ligand data:

- rigid body docking (eg. *DiffDock*, *EquiBind*)
- flexible pocket docking or co-folding (eg. *NeuraLPlexer*, *Rosetta-Fold All-Atom*, *UMol*, *AlphaFold3*, *HelixFold*, *Chai-1*, etc.)
- pocket-conditioned ligand generation or optimization (eg. *DiffSBDD*, *DiffLinker*)
- ligand-conditioned protein engineering (eg. *LigandMPNN*)

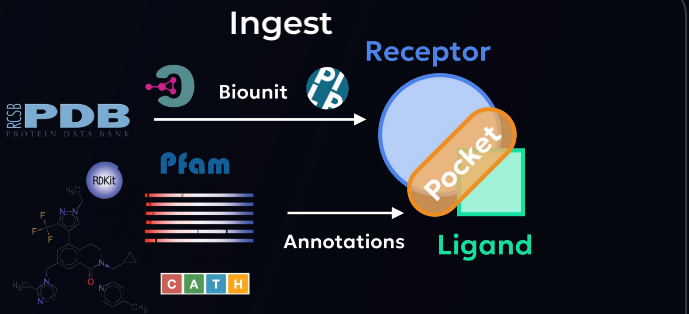
What ML field needs for further progress:

- Training set diversity
- Low information leakage
- Test set quality
- Test set diversity
- Realistic inference scenarios

- Clear evaluation of SOTA (metrics and method Leaderboard)

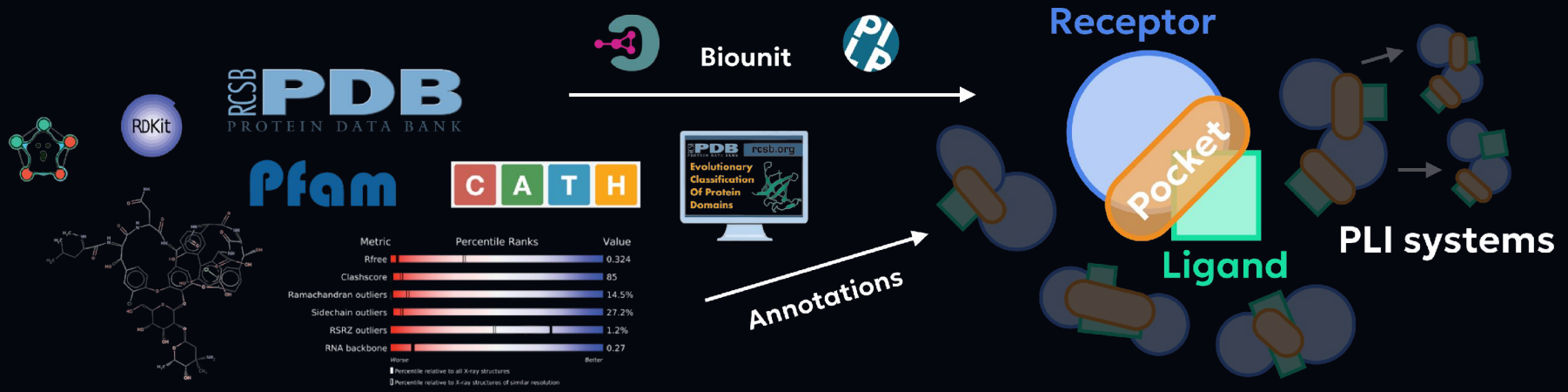
* list is not exhaustive

How it’s made? Let’s begin from the start!



- Collect **all** PLIs in the PDB
- **Pair** *apo* and AF2 structures
- Annotate **everything**
- Keep up with PDB **updates**

Collect and annotate all PLIs



- PDB NextGen Archive to fetch and generate all biological assemblies
- PLI system is a combination of nearby ligand and protein chains
- Clean files: CIF/PDB files of proteins and (fixed bonding) SDF files of ligands
- **Protein** properties also aggregated to the **pocket**
- **Ligand** properties with many molecular characteristics
- Around 500 annotations for each PLI system

Making data usable: overcoming ingest issues

B. Dataset

Preprocessing. The time split is done after preprocessing the 19 443 complexes of PDBBind v2020 as follows. First, we drop all complexes that cannot be processed by the RDKit library (Landrum, 2016), leaving 19 119 complexes. We process each ligand and receptor with OpenBabel (Open Babel development team, 2005). Next we correct all receptor hydrogens and add missing ones using `reduce`[4].

We used PDBbind from 2019 (2020 release[29]) processed by the authors from EquiBind (https://zenodo.org/record/6408497, 19119 protein-ligand complexes). We parsed all protein sequences from the PDB files. 18884 out of 19119 protein structures (99%) could be parsed (<80% missing CAs and >50 residues). Only the first protein chain in all protein-ligand complexes used here and in the evaluation was extracted. Features (see below) could be generated for 17936/18884 (95%) protein-ligand complexes. The failed ones did so due to issues of converting SMILES to 3D structures using RDKit (version 2023.03.2, https://www.rdkit.org).

Recent publications have dropped nearly 10% of initial dataset just because of ingest issues.

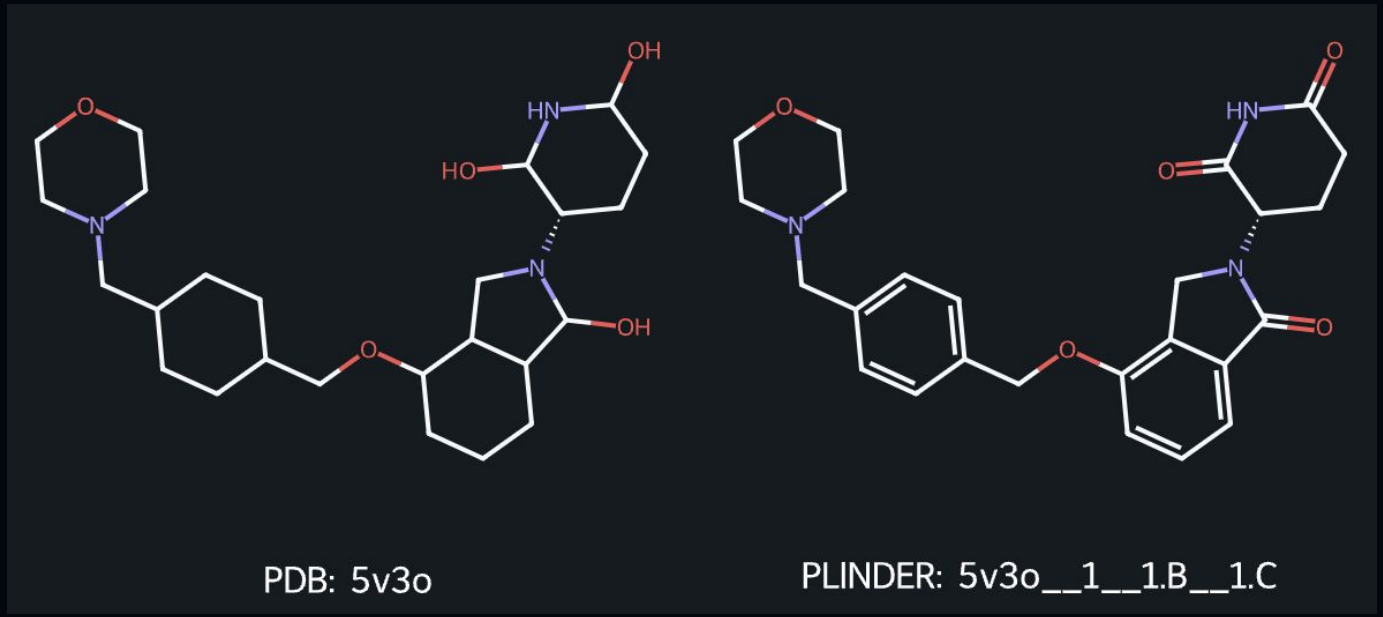
As many as 1272 are attributed to failures to RDKit processing.

All PLINDER ligands can be loaded into RDKit, successfully saved to SDF and reloaded !

Fixing Ligand bonding using RDKit

We use SMILES from Chemical Component Dictionary to assign bond orders

- AllChem.AssignBondOrdersFromTemplate(template, mol)



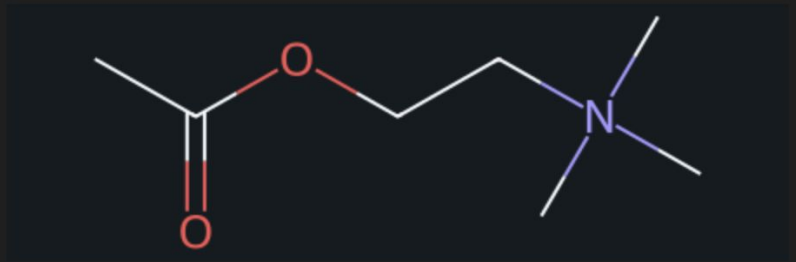
Keeping “sanity” with poorly defined SMILES

Not all SMILES are acceptable to RDKit Sanitization.

```
# AtomValenceException
mol = Chem.MolFromSmiles("CC(=O)OCCN(C)(C)C", sanitize=False)
problems = Chem.DetectChemistryProblems(mol)
mol
```

✓ 0.0s

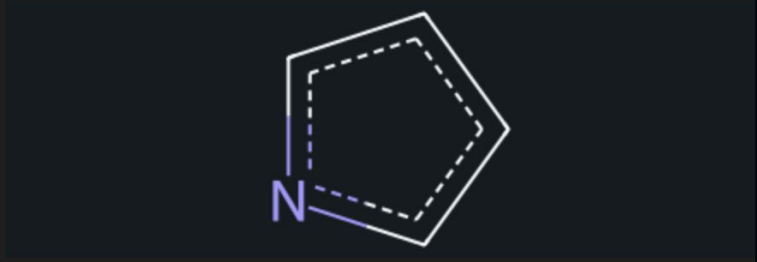
```
[14:57:27] Explicit valence for atom # 6 N, 4, is greater than permitted
```




```
# KekulizeException
mol = Chem.MolFromSmiles("c1ccnc1", sanitize=False)
problems = Chem.DetectChemistryProblems(mol)
mol
```

✓ 0.0s

```
[14:57:27] Can't kekulize mol.  Unkekulized atoms: 0 1 2 3 4
```



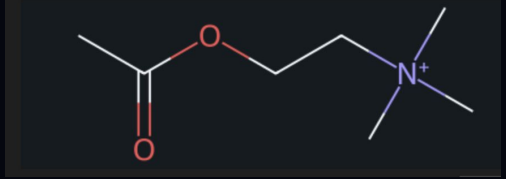
Keeping “sanity” with poorly defined SMILES

Good news: We seem to fix that!
Other news: might need to fix it again as it relies on SetNumExplicitHs / GetExplicitValence

```
160         # deal with remainng issues, if any
161         for p in ps:
162             if p.GetType() == "AtomValenceException":
163                 at = mol.GetAtomWithIdx(p.GetAtomIdx())
164                 atm_no = at.GetAtomicNum()
165                 formal_charge = at.GetFormalCharge()
166                 valency = at.GetExplicitValence()
167                 elem_max_explicit_valency = max_explicit_valency_per_element[atm_no]
168                 expected_charge = valency - elem_max_explicit_valency
169                 if expected_charge > formal_charge:
170                     # Fix Explicit valence issue
171                     at.SetFormalCharge(expected_charge)
172             if p.GetType() == "KekulizeException":
173                 # hack: only works for nitrogens with missing explicit Hs
174                 for atidx in p.GetAtomIndices():
175                     at = mol.GetAtomWithIdx(atidx)
176                     # set one of the nitrogens with two bonds in a ring system as "[nH]"
177                     if at.GetAtomicNum() == 7 and at.GetDegree() == 2:
178                         at.SetNumExplicitHs(1)
179                         break
180         sanitize_mol(mol)
181         return mol
```

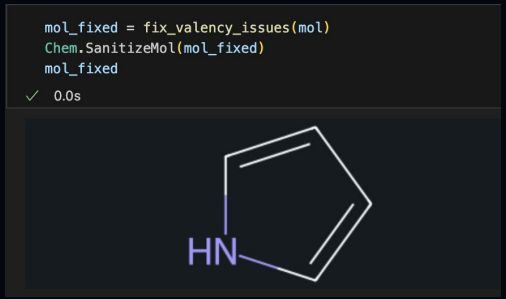
```
mol_fixed = fix_valency_issues(mol)
Chem.SanitizeMol(mol_fixed)
mol_fixed
```

✓ 0.0s



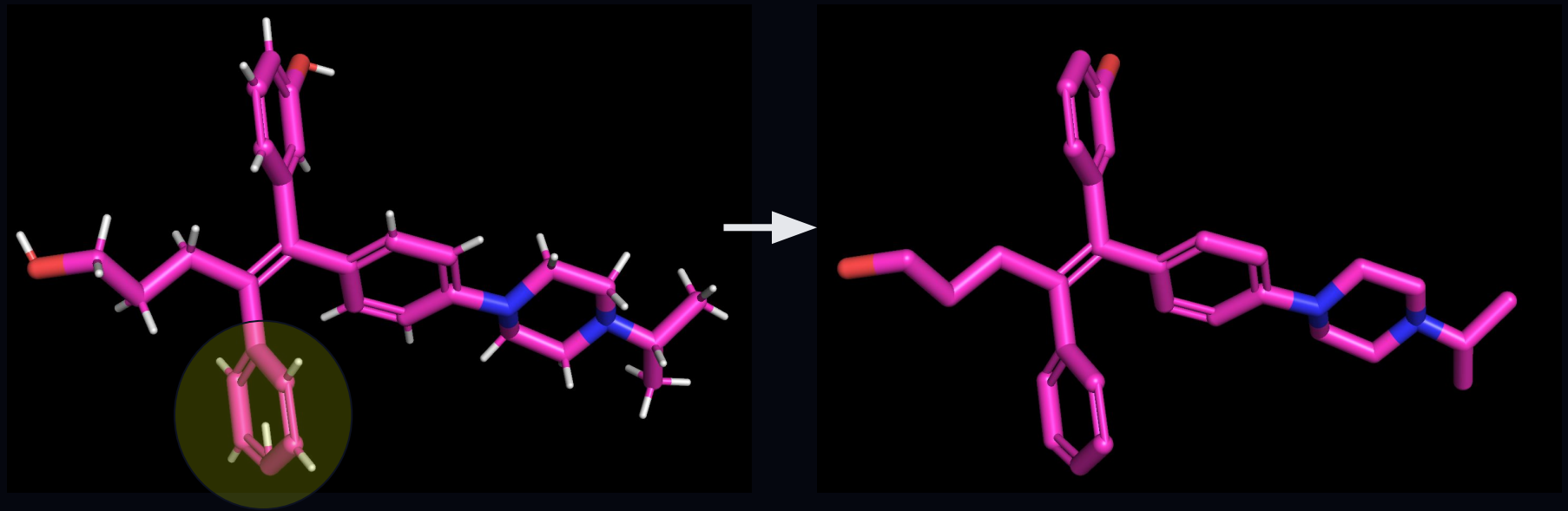

```
mol_fixed = fix_valency_issues(mol)
Chem.SanitizeMol(mol_fixed)
mol_fixed
```

✓ 0.0s



To standardize we remove hydrogens

Hydrogens present in the models are too often modelled without experimental observations, not consistently added, or straightforwardly nonsensical (eg. 6a6k)



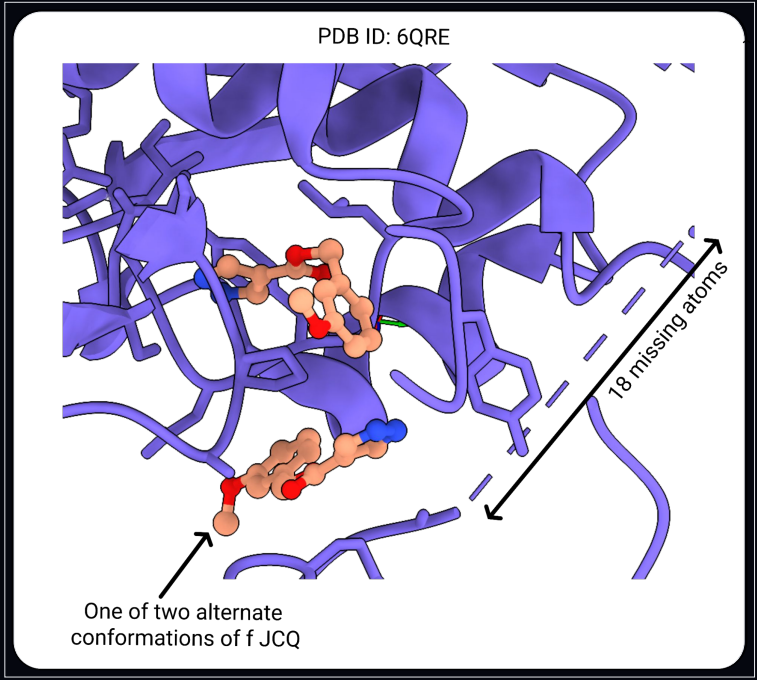
Why quality annotations matter?

Experimental structure quality varies a lot!

- uncertainty due to missing density
  - unresolved ligand atoms
  - missing residues
- vdW clashes in protein pocket
- alternative residue locations
- crystal lattice contacts
- etc.

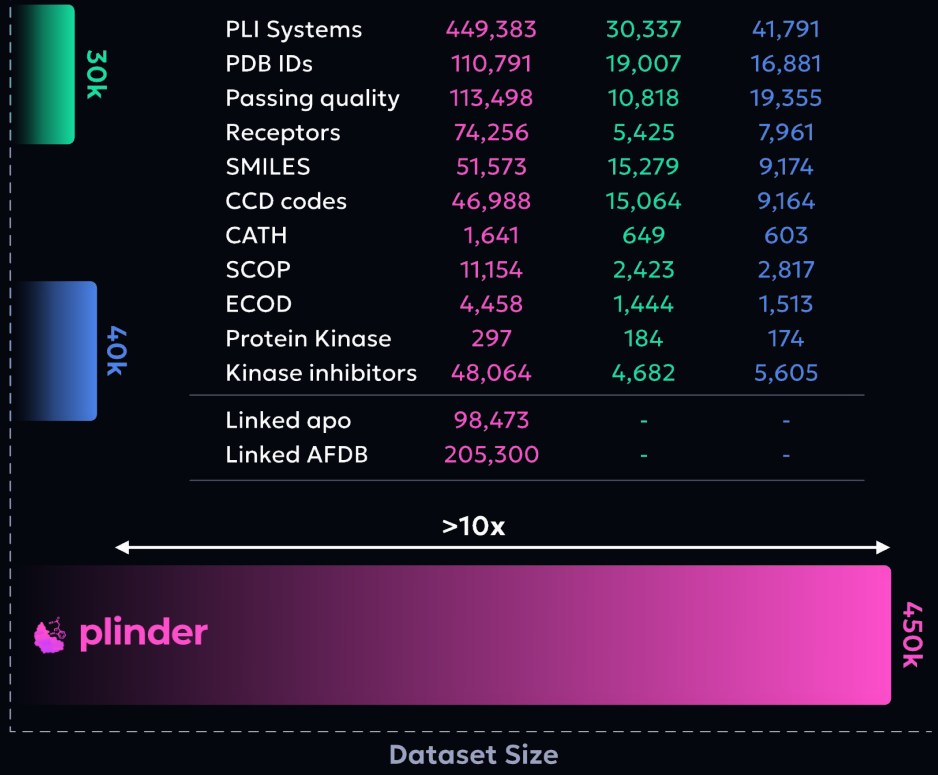
Less than 50% of PDBBind structures pass all our “high quality” criteria

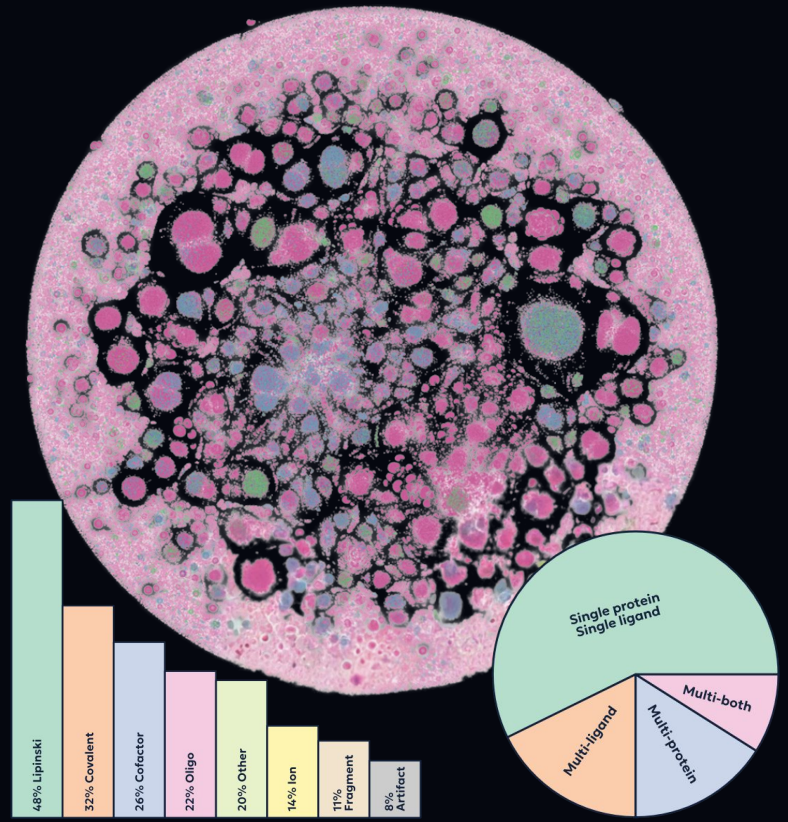
Ambiguous “ground truths” make tests unreliable!
→ Evaluation sets need high quality structures



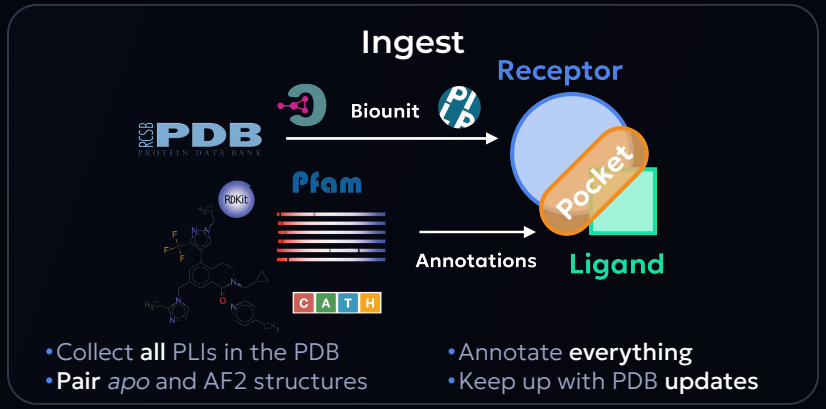
PLINDER expands the dataset size more than 10x !

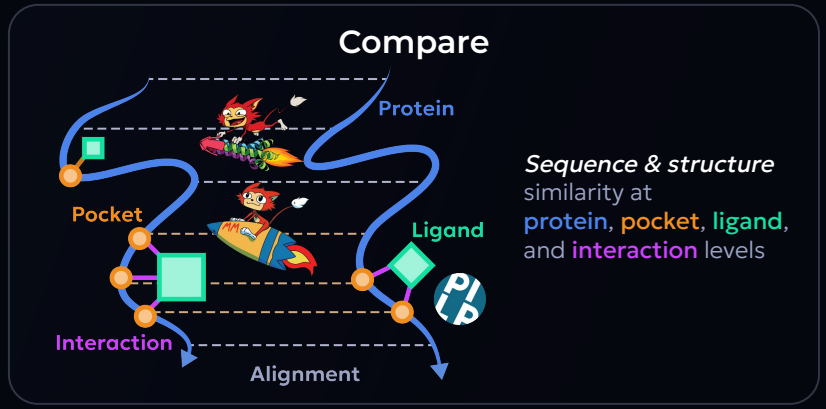
	plinder	PDBBind	DockGen
PLI Systems	449,383	30,337	41,791
PDB IDs	110,791	19,007	16,881
Passing quality	113,498	10,818	19,355
Receptors	74,256	5,425	7,961
SMILES	51,573	15,279	9,174
CCD codes	46,988	15,064	9,164
CATH	1,641	649	603
SCOP	11,154	2,423	2,817
ECOD	4,458	1,444	1,513
Protein Kinase	297	184	174
Kinase inhibitors	48,064	4,682	5,605
Linked apo	98,473	-	-
Linked AFDB	205,300	-	-





Automated and reproducible end-to-end workflow



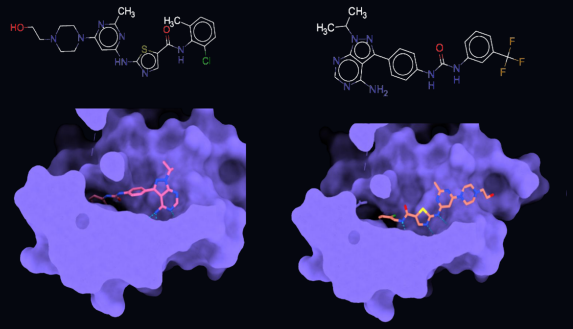


Which PLI systems are similar?

Same protein
Same pocket
Different ligand

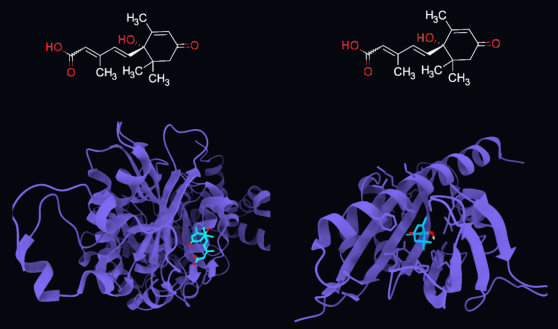
A. Same protein, same pocket, different ligand

PDBID: 3QLG PDBID: 3QLGF



B. Different protein, different pocket, same ligand

PDBID: 4MKV PDBID: 7Z1R

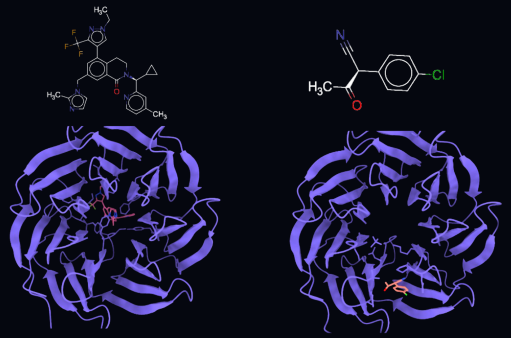


Different protein
Different pocket
Same ligand

Same protein
Different pocket
Different ligand

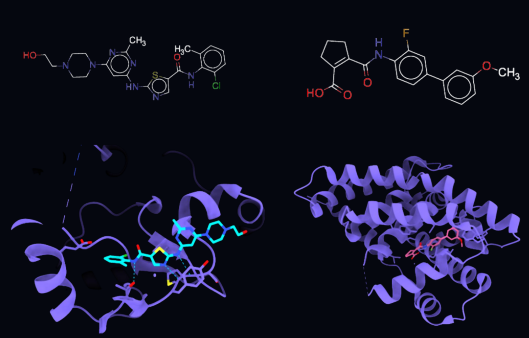
C. Same protein, different pocket, different ligand

PDBID: 7UAS PDBID: 6UJH



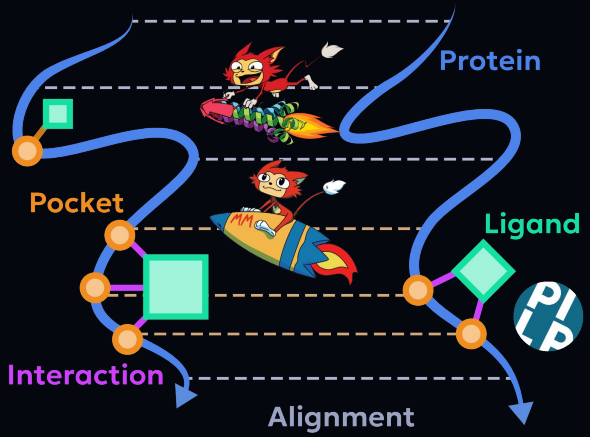
D. Different protein, different pocket, different ligand

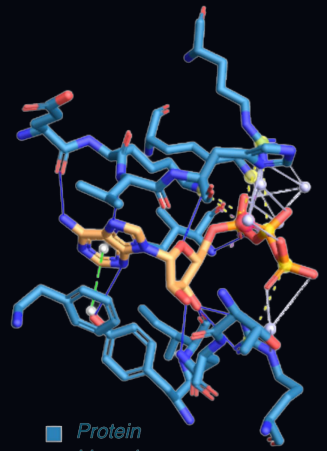
PDBID: 3QLG PDBID: 5Y1J



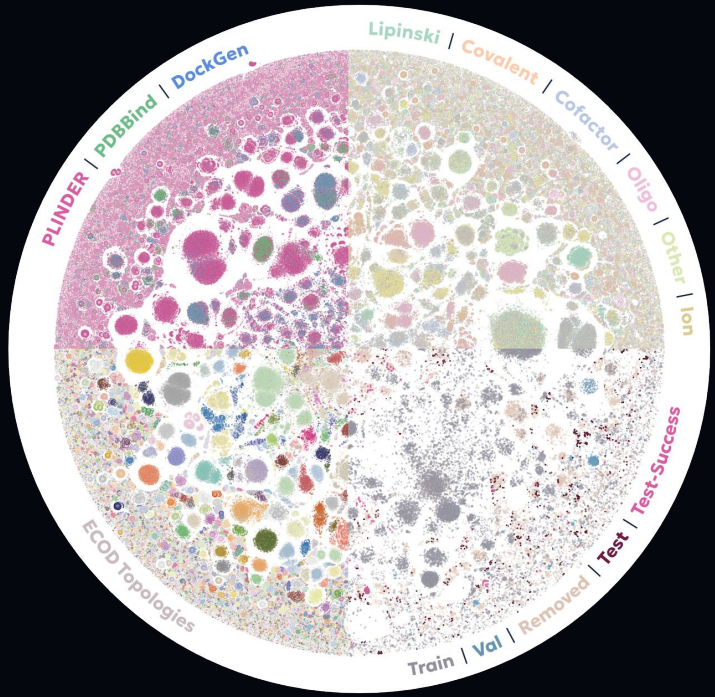
Different all

Measure similarity between PLIs





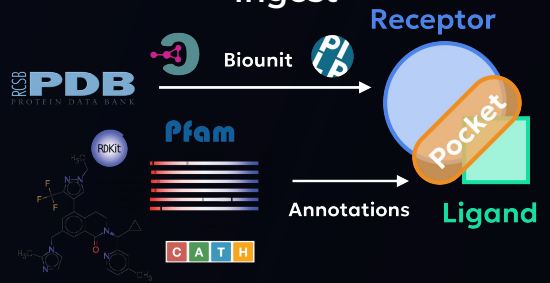
Protein
Ligand
Water
Charge Center
Aromatic Ring Center
Metal Ion
Hydrophobic Interaction
Hydrogen Bond
Water Bridge
π-Stacking (parallel)
π-Stacking (perpendicular)
π-Cation Interaction
Halogen Bond
Salt Bridge
Metal Complexation



- **All vs. all** Foldseek/MMseqs on **protein** chains
- Alignments → **pocket** & **interaction** similarities
- ECFP4 fingerprints for **ligand** similarities
- 14 similarity metrics and >20B scores
- **Graph** based community **clustering**

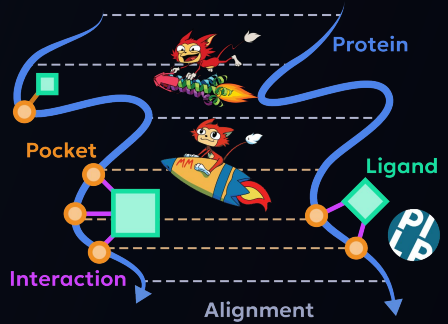
Automated and reproducible end-to-end workflow

Ingest



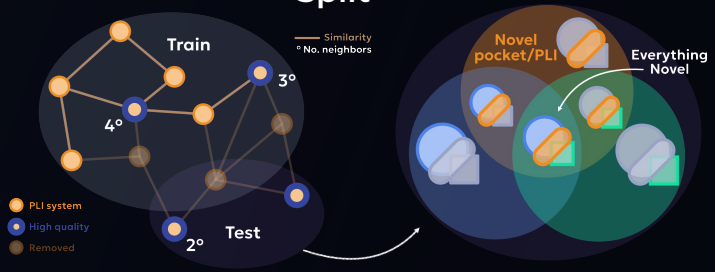
- Collect **all** PLIs in the PDB
- **Pair** *apo* and AF2 structures
- Annotate **everything**
- Keep up with PDB **updates**

Compare



Sequence & structure similarity at **protein**, **pocket**, **ligand**, and **interaction** levels

Split



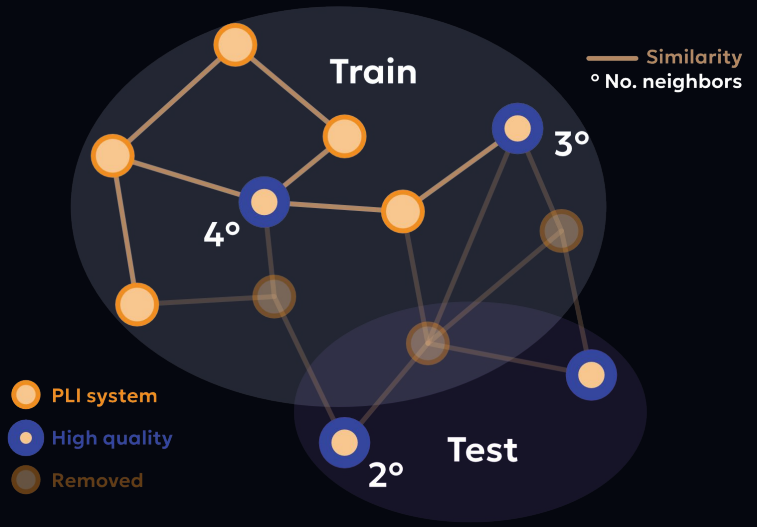
Maximizing test set **quality and diversity** while ***minimizing*** information **leakage**

Stratification for different tasks

Novel algorithm for optimal train/val/test splits

```
function SPLITTING(S, C, G, m, M)
    proto_test ← ∅
    for s ∈ S do
        if pass_quality (s) then
            N_s ← ∅
            for g ∈ [1, |G|] do
                leaked ← neighbors (s, G_g)
                N_s ← N_s ∪ leaked
            if m < |N_s| < M then
                proto_test.insert (s)
    ▷ Sort s ∈ proto_test by |N_s|
    test ← ∅
    for c ∈ C do
        test ← test ∪ upto 1 from proto_test for c
    train ← S
    for s ∈ test do
        train ← train \ N_s
```

Systems
Clusters
Graphs
min & Max. removal



Optimize for:

- Test set **quality**
- Test set **diversity**
- Minimal information **leakage**
- Training set **size** and **diversity**

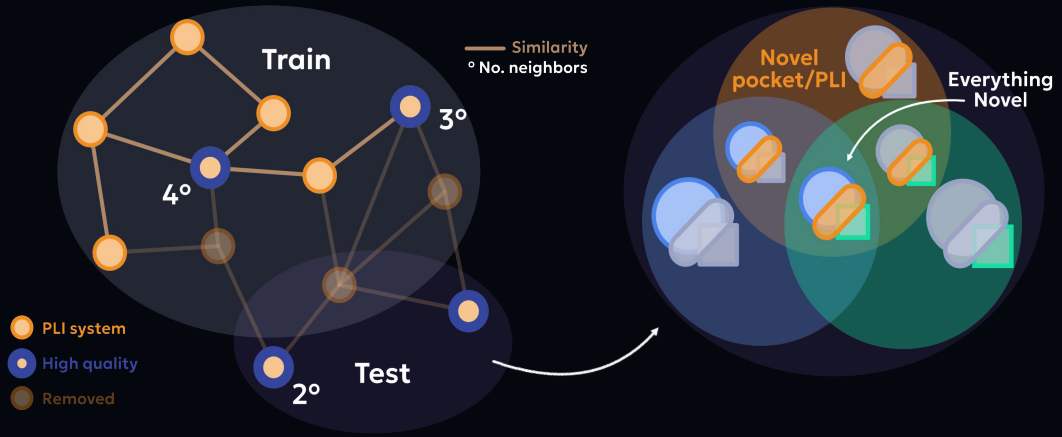
PLINDER split v2 (released 2024-08-19)

Split systems by multigraph (depth of 1)

- **protein** sequence similarity at **30%** (note: similarity not identity)
- unique **PLI** similarity at **30%**

split	No. of Systems	PDB IDs	Unique CCD codes	PLI communities	Pocket communities	Ligand communities
test	1036	1020	689	500	483	336
train	309140	76901	36054	29701	9408	8433
val	832	573	318	314	281	87
removed	98718	34181	15262	7801	3160	3281

Splits to evaluate method generalization!



Stratify to:

- Novel **protein**
- Novel **ligand**
- Novel **pocket**/**Xns**
- **Everything** novel

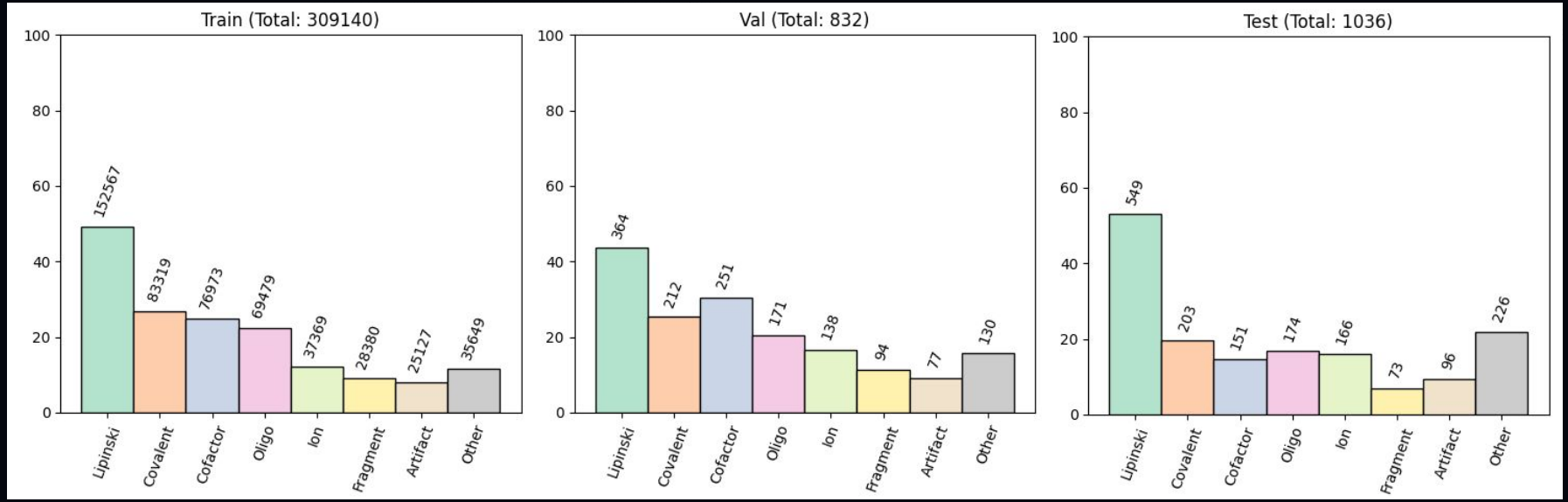
split comparison	test vs train	test vs val	val vs train
total	1036	1036	832

Novelty in split

Protein	1030	1033	192
Ligand	68	349	13
Pocket & PLI	1035	1036	812
All of above	67	347	0
None	1	0	20

Note: **val vs train** is not as strictly deleaked, but there is a good feature (see eval part)

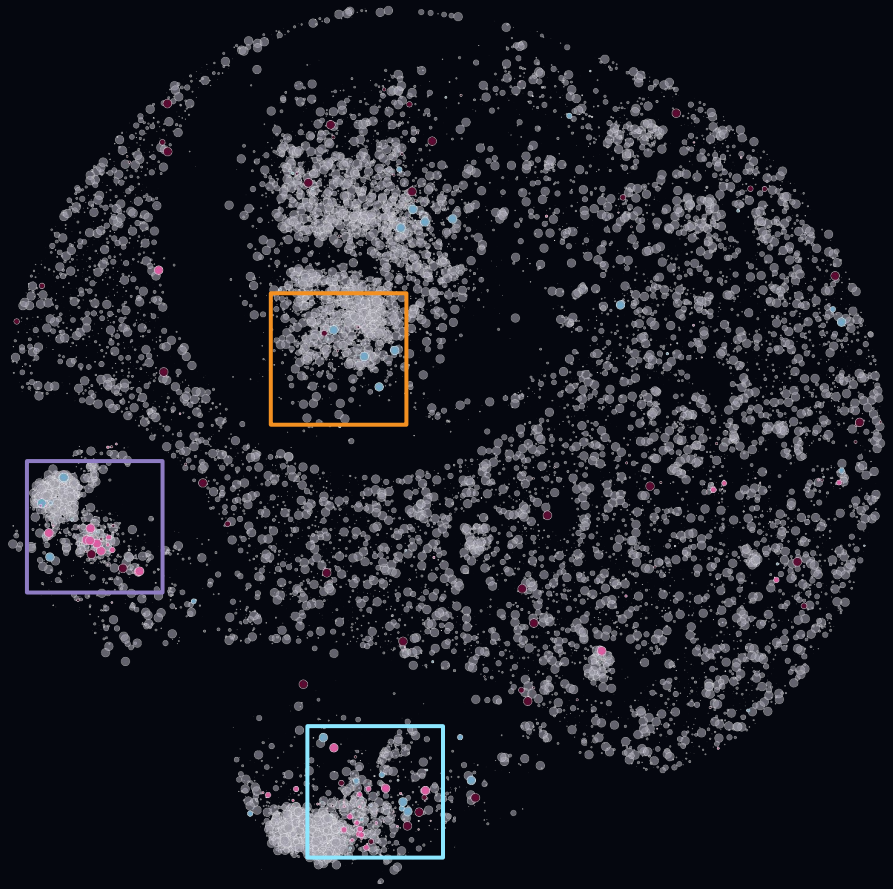
Is test representative for ligand types?

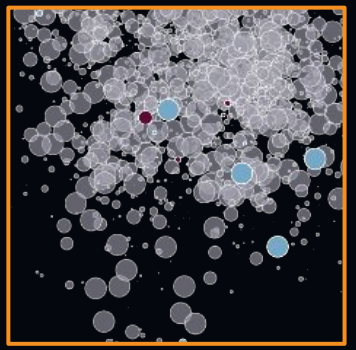


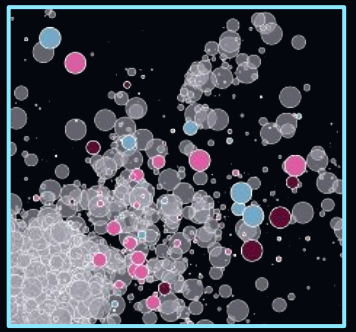
Test set of high ligand diversity and some novelty

Ligand cosmograph clustered by ECFP4 Tanimoto sim > 0.5 (scaled by size)

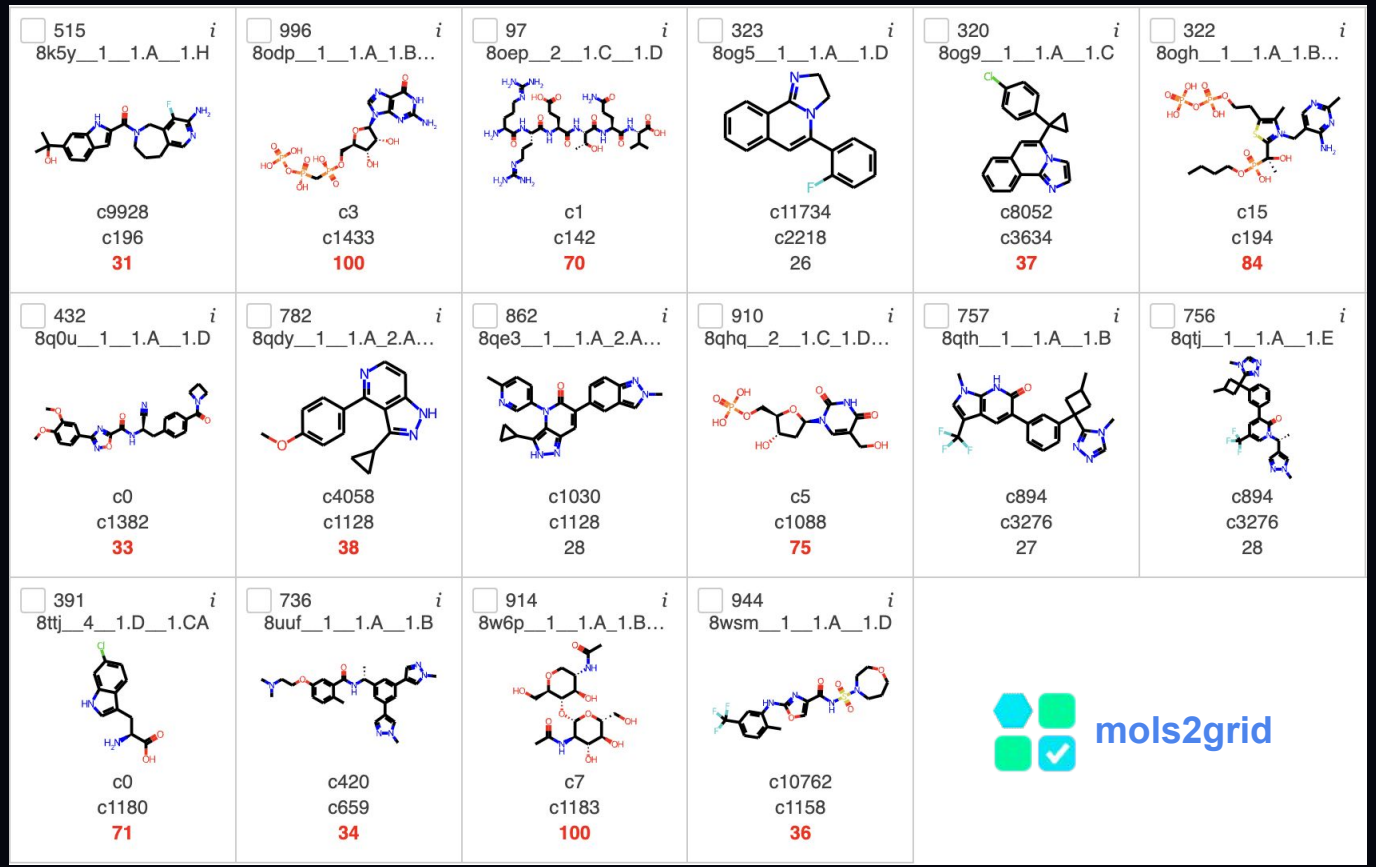
- **Train only**
- **Val only or Train/Val**
- **Test only**
- **Train/Test or Val/Test**







Snapshot of ligands in test

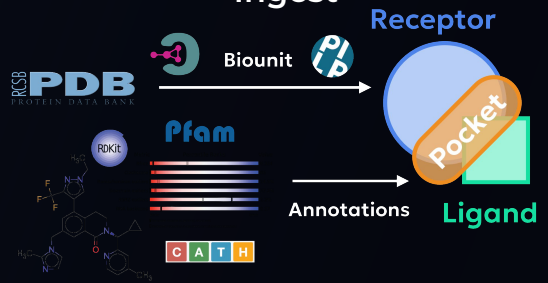


System ID

Ligand community cluster id
PLI community cluster id
Ligand similarity to train

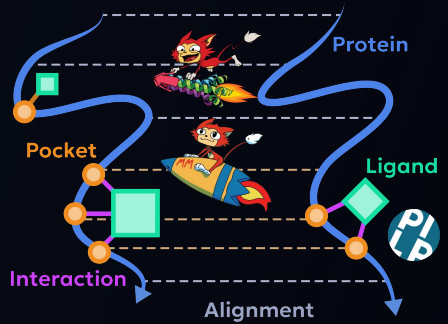
Automated and reproducible end-to-end workflow

Ingest



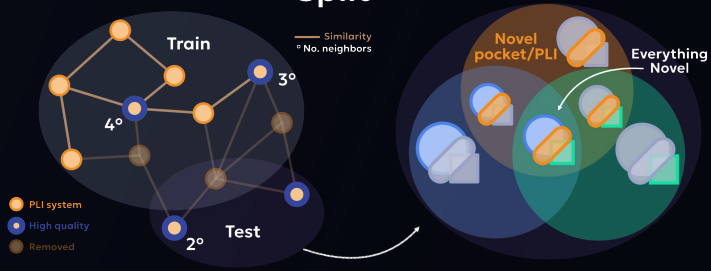
- Collect **all** PLIs in the PDB
- **Pair** *apo* and AF2 structures
- Annotate **everything**
- Keep up with PDB **updates**

Compare



Sequence & structure similarity at **protein**, **pocket**, **ligand**, and **interaction** levels

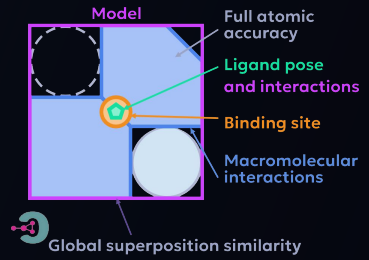
Split



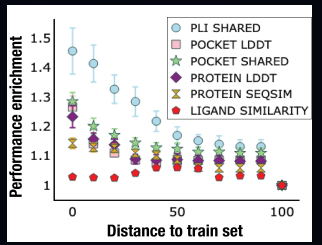
Maximizing test set **quality and diversity** while ***minimizing*** information **leakage**

Stratification for different tasks

Evaluate



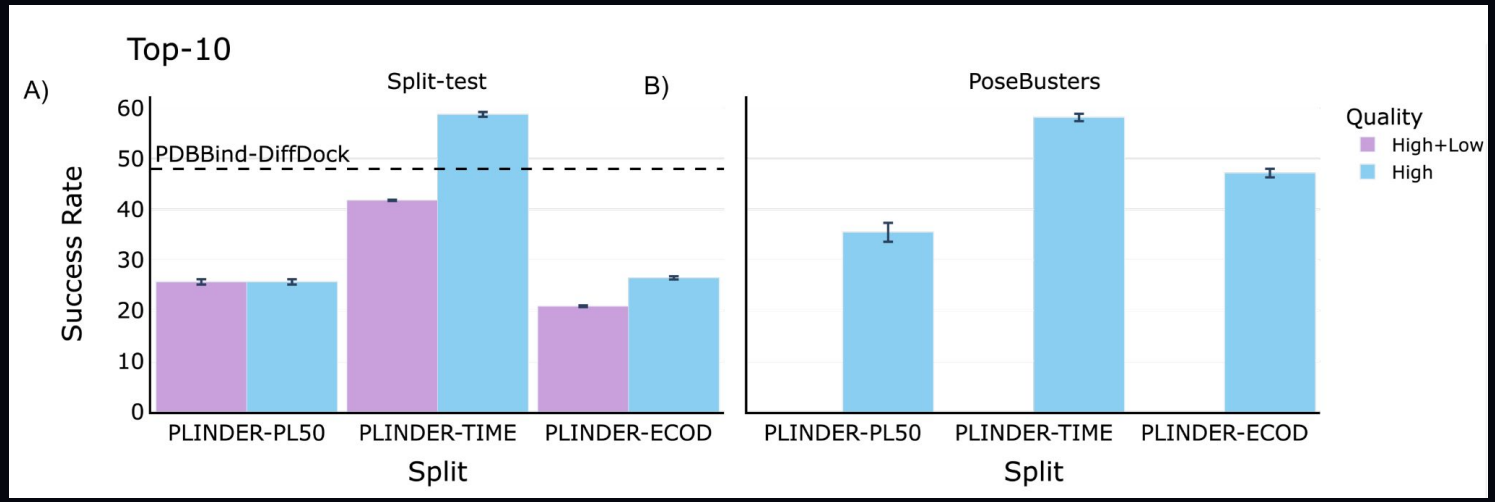
Comprehensive **evaluation** harness



Resources to **inspect biases** in your models

PLINDER provides a trainable dataset (earlier iteration)

In collaboration with NVIDIA BioNeMo team we have re-trained DiffDock on PLINDER (**v0**).
Using three splitting strategies we evaluated against two test sets:
1) test set as given by split strategy and 2) (non-deleaked) PoseBusters set



Success rate defined by fraction with < 2 A RMSD best pose cutoff

NVIDIA

System similarity scores give insights into evaluation

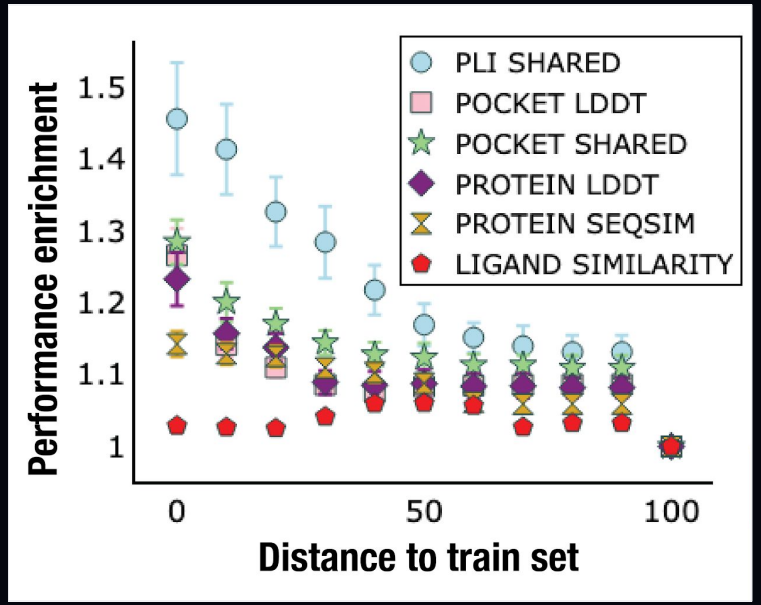
Retrained DiffDock results for PoseBusters set were evaluated as a function of similarity distance

Some observations:

- Binding site interaction information (probed by **PLI shared** distance) are most sensitive
- This is followed by pocket sequence (**Pocket shared**) and shape (**Pocket LDDT**) similarities
- These seem to agree with the intuition for rigid docking task

Caution notes:

- likely different for other tasks or model
- could change if “success” metric was redefined

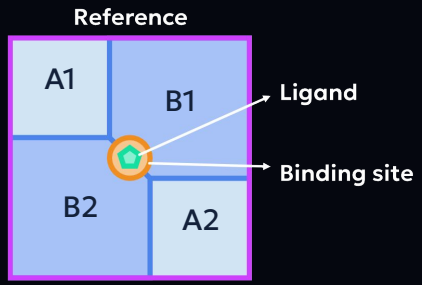


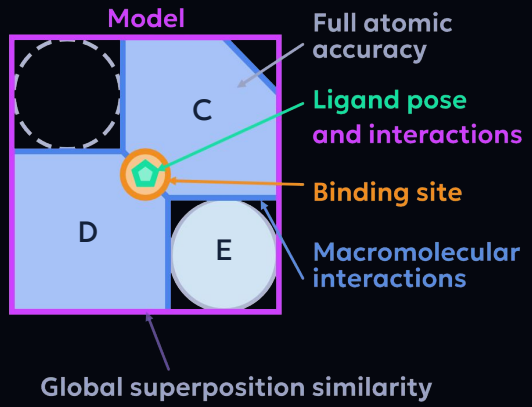
NVIDIA

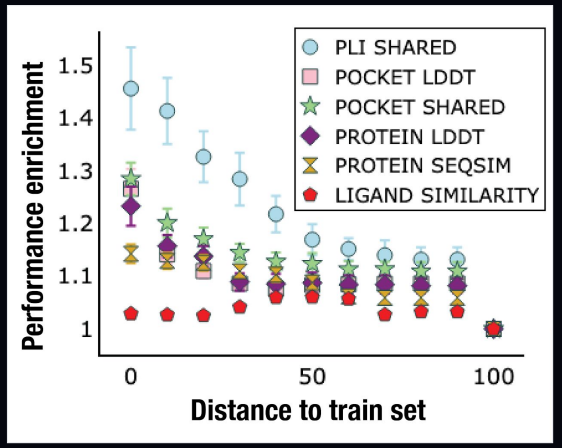
Evaluate performance and its determinants

Comprehensive eval harness

- Go **beyond rigid placement** of ligand <5A RMSD for **single** ligands
- Check how well you're predicting **ligand** poses and physics, **pocket** residues and **interactions**
- Covers the whole **range of metrics** used in **CASP**







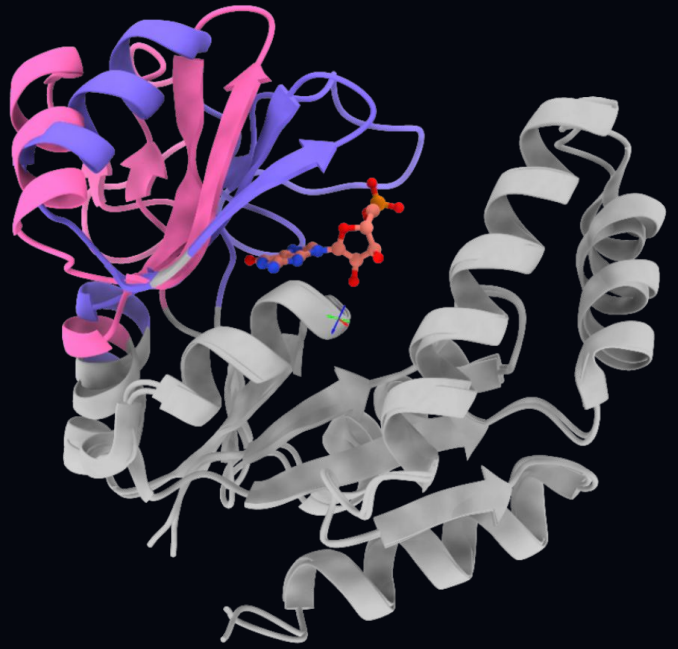
Learn what's being learned

- Look at how prediction **performance changes with leakage**
- Understand the **inductive biases** in your model
- Explore options to **improve** what your model learns

Moving towards realistic inference scenarios

To establish SOTA we need realistic inference scenarios

- Previous benchmarks focused on "re-docking"
- Start your inference from **apo or predicted** structures as your users would!
- **Augment training** with these structures to teach your models what to do with them



PDBID: 1EX7: Bound (holo) structure

PDBID: 1EX6: Unbound (apo) structure

What is next for PLINDER?

More data

- Measured & predicted binding affinity and other interaction metrics
- Conformational changes: crypticity annotations of binding pockets
- Adding predicted structures for every holo structure to unlock flexible structure prediction at full PLINDER scale
- Data augmentation strategies incl. holo minimization to expand apo coverage, Van-Der-Mers & Cross-docking, etc.

Updates

- Regular update cycles with new PDB NextGen Archive releases

Better UX

- DataLoader: direct access from dataset to tensors
- Improved multi-protein, multi-ligand and covalent ligand support

Leaderboard

- Comparison of open-sourced models trained and evaluated on PLINDER
- Inference models accessible to the community for testing (Hugging Face)
- Establish baseline performance of physics-based methods

Collaborations

- We remain open to community contributions and larger collaborations

MLSB 2024 Challenge

Dates

- Sep 24: Detailed instructions (during training workshop)
- Oct 9: Leaderboard opens
- Nov 9: Leaderboard closes
- Nov 27: Winners notified
- Dec 14/15: MLSB workshop

Sign up here

Challenge Overview

- Focus on flexible docking and co-folding predictions
- Utilizing provided datasets from PLINDER and PINDER (for protein-protein interactions)
- Standardized submission system based on HuggingFace Spaces
- For more details see: mlsb.io/#challenge

Organizers

VantAI · NVIDIA · BIOZENTRUM Universität Basel The Center for Molecular Life Sciences · MLSB

Source: PINDER: The protein interaction dataset and evaluation resource, biorxiv (2024)

Moving Beyond Memorisation

Resources for Generalisable Protein Complex Prediction

Online Training Workshop

When: 24 September 2024, 3pm CEST

Register here: plinder.sh/blog/training

Schedule:

Time (in CET)	Session
15:00 – 15:20	Welcome and Introduction
15:20 – 15:50	Metrics for measuring protein complex model accuracy
15:50 – 16:40	Generalisation, augmentation, and inference scenarios
16:40 – 17:00	Break
17:00 – 18:30	Parallel sessions on PINDER and PLINDER
18:30 – 19:00	Break
19:00 – 20:00	Huggingface model and leaderboard submission
20:00 – 20:30	Next steps

What to expect:

- Hands-on exercises on effectively using **large-scale datasets** for training your models.
- Strategies to **boost your model's generalisation performance** by preventing task-specific information leakage and employing advanced augmentation techniques.
- Insights into key **metrics for evaluating the accuracy** of structure predictions.
- Practical experience with realistic inference scenarios and **leaderboard submissions**.

VantAI

Acknowledgements

VantAI
Yusuf Adeshina
Vladas Oleinikovas
Thomas Duignan
Xuejin Zhang
Daniel Kovtun
Emanuele Rossi
Clemens Isert
Mehmet Akdel
Zachary Carpenter
Michael Bronstein
Luca Naef

MIT CSAIL
Gabriele Corso
Hannes Stärk

SIB/UniBas
Janani Durairaj
Xavier Robin
Gabriel Studer
Gerardo Tauriello
Torsten Schwede

NVIDIA
Zhonglin Cao
Zachary McClure
Guoqing Zhou
Srimukh Prasad
Veccham
Yuxing Peng
Prabindh Sundareson
Emine Kucukbenli

Core PLINDER team

Questions?

gs://plinder

github.com/plinder-org

plinder.sh

MLSB 2024: PLINDER challenge

plinder

When

- Leaderboard opens: Oct 9
- Leaderboard closes: Nov 9

Train/val/test sets

- as specified by PLINDER

Joint protein-ligand complex prediction

- Single ligand, single protein
- Flexible/co-folding task
- Accuracy metric: lDDT-PLI

Model submission

- with HuggingFace spaces
- Each assigned a GPU (T4, A10G, A100)
- Limited period of time to run inference

Pre-requisites

- MLSB submission + open-source code + train.py script

Inference

- sequence (or PLINDER-provided apo) and SMILES

VANTAI

PDBBind

~30k systems
19k PDB IDs
15k unique ligands
>40% don't pass quality criteria
splits not tailored for ML tasks

PLINDER

~450k systems
111k PDB IDs
52k unique ligands
>100k pass quality criteria
*Diverse and high quality **splits** with low leakage*
*> 20B Protein, pocket, PLI and ligand **similarities***
***Automated** to run on new PDB releases*
*Linked to **apo and predicted** structures for input augmentation and realistic inference scenarios*