KATANA GRAPH

June 08, 2022

Katana @ Cambridge Cheminformatics Meeting

Cheminformatics At Scale On a Graph Platform

Andrew Stolman

Senior Software Engineer

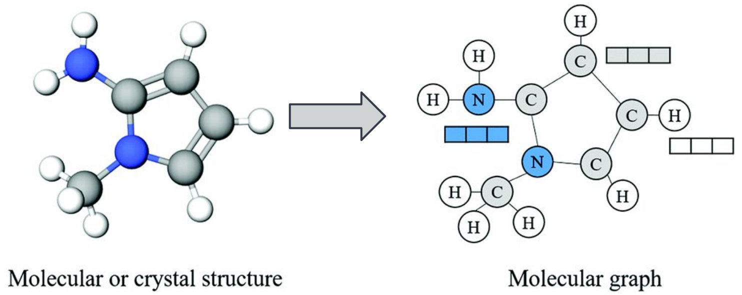
astolman@katanagraph.com

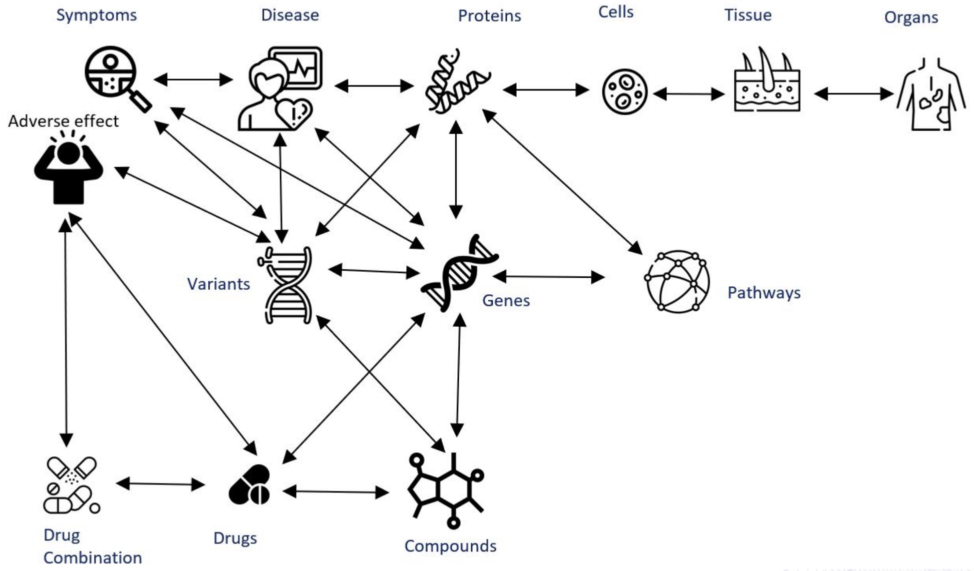
Special thanks to

- Bo Wu, Suman Bera, Thomas Cook, Kamesh Peri, Chris Gessner, Gurbinder Gill and everyone at Katana
- Abhik Seal, Brian Martin, Phillip J Hajduk, Jennifer Van Camp and our collaborators at Abbvie

Introduction: what are graphs?

- Graphs are abstract objects which represent relationships among entities made up of **nodes** and **edges**
- Graphs can represent **heterogeneous** data and complex relationships

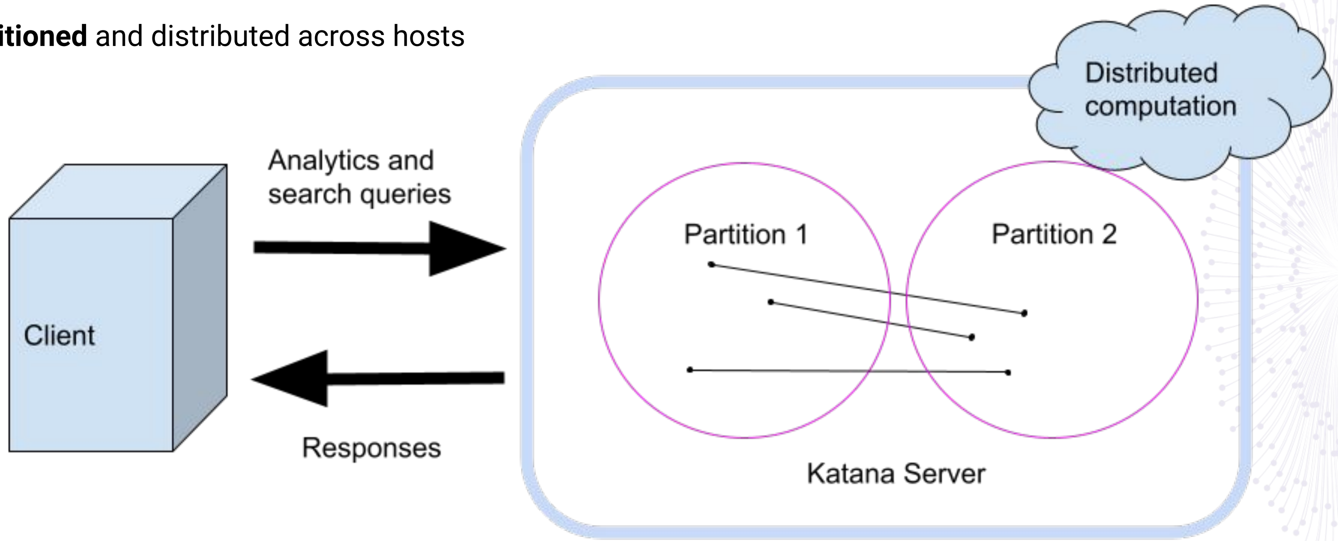




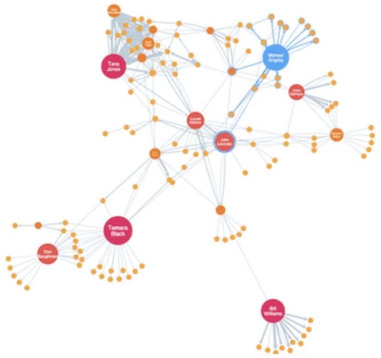
Distributed graph platforms

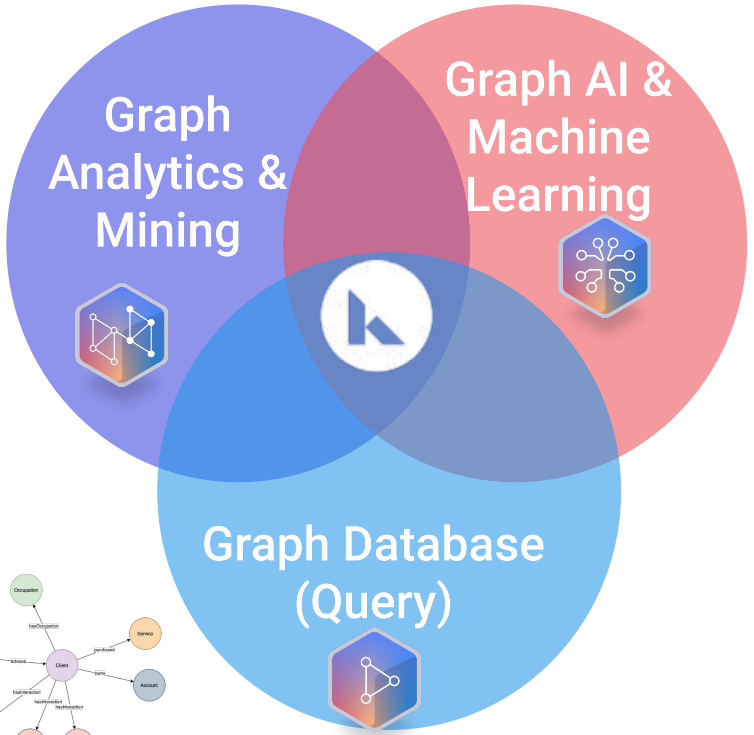
Distributed graph platforms are for handling millions and billions of nodes and edges

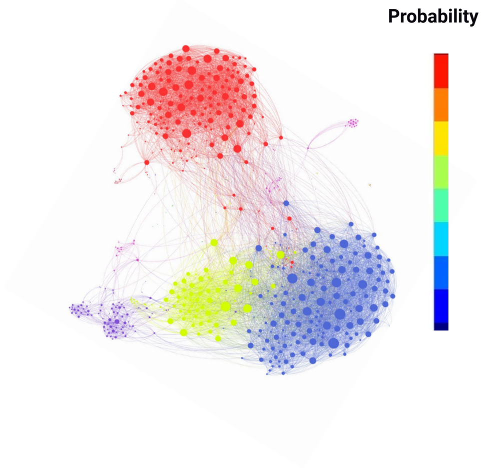
Graphs are **partitioned** and distributed across hosts

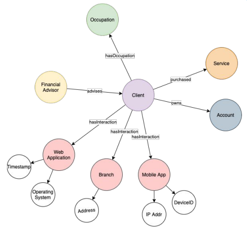


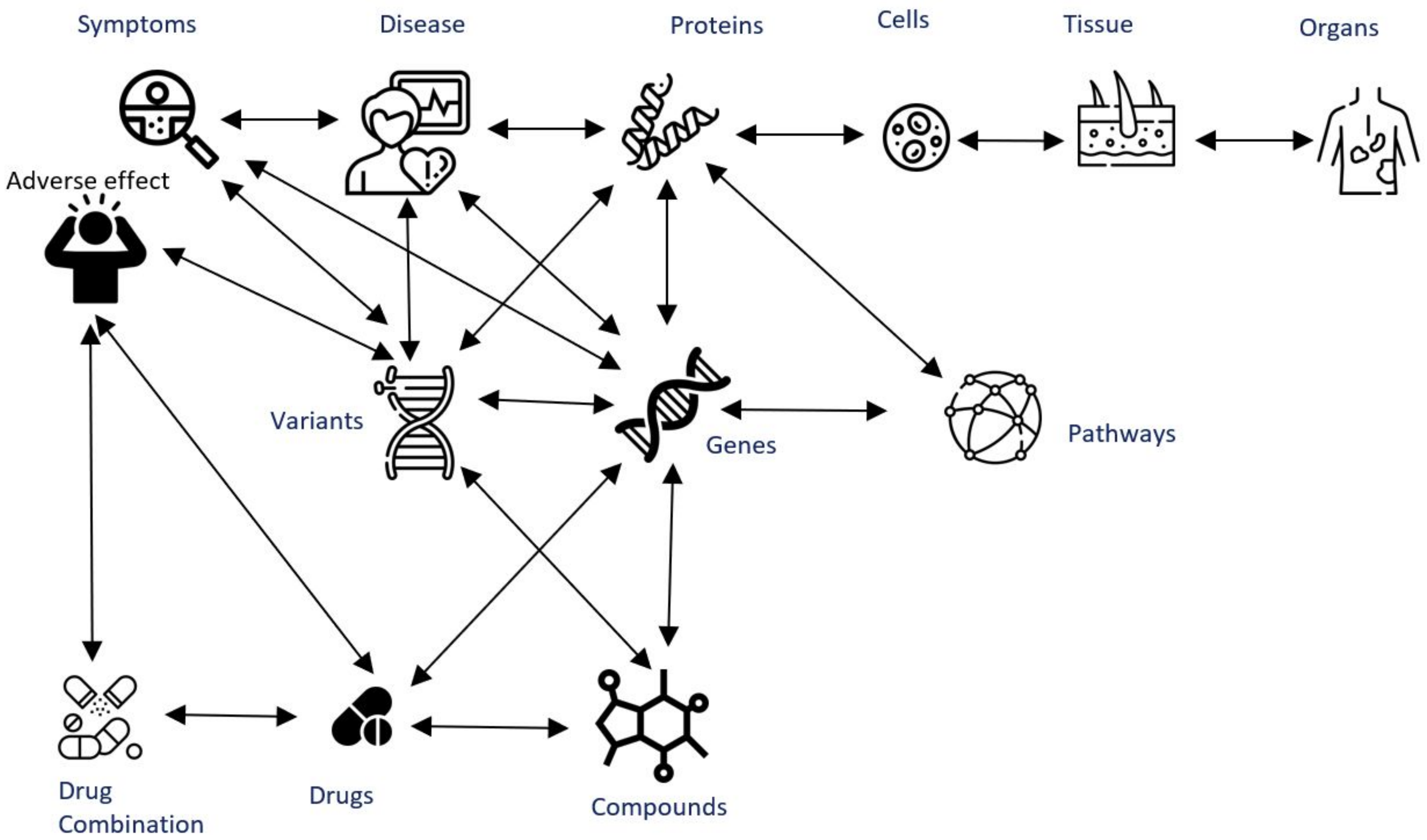
Graph Compute Domains





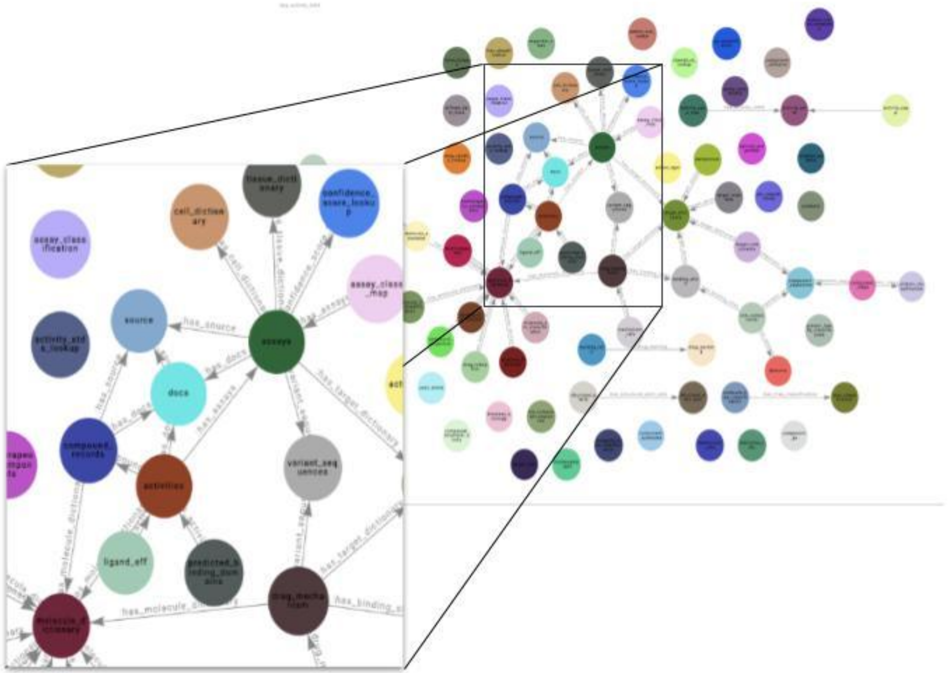





Symptoms
Disease
Proteins
Cells
Tissue
Organs
Adverse effect
Variants
Genes
Pathways
Drug Combination
Drugs
Compounds

ChEMBL 30 Graph Schema

- 2.3M nodes
- 37.3M edges
- 76 node types
- 49 relation types



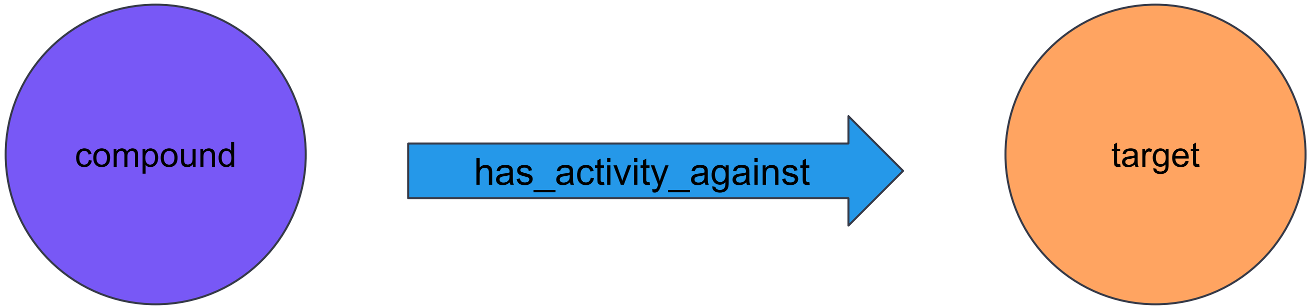
OpenCypher introduction

- OpenCypher is the most popular open source graph query language
- A query language like SQL but customised to search graph patterns
- Example:

  -
    ```
    MATCH (a:compound)-[r:has_activity_against]-(b:target{name:”target”}) RETURN count(r)
    ```

Finds all compounds which are known to have activity against a given target

Count all the relationships found



Katana OpenCypher rdkit integration

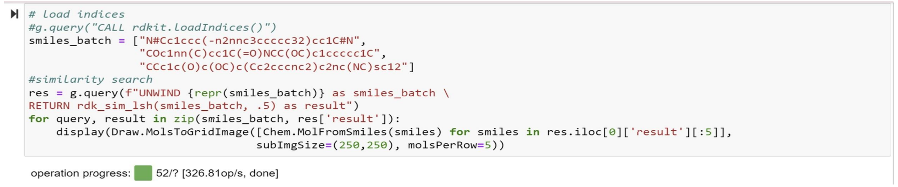
- We extended OpenCypher with the following functions:
  - Find minimum common substructure
  - RDK fingerprint
  - Morgan fingerprint
  - Topological torsion fingerprint
  - Erg fingerprint
  - Tanimoto similarity
  - Similarity search
  - Substructure search

Similarity indexing

Katana 24x faster than PostGres

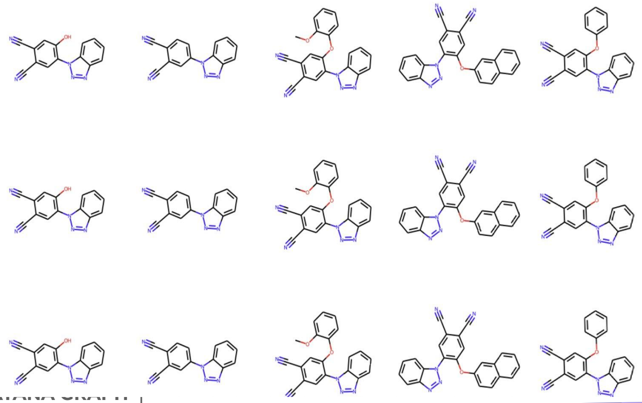
For similarity search indexing we use **distributed minhash LSH**

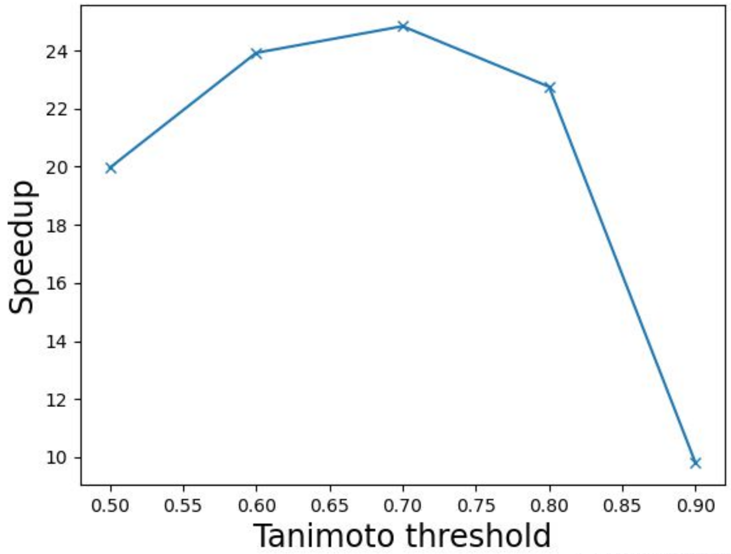
Benchmarks were run on a single machine with 1K random query molecules



```
# load indices
#g.query("CALL rdkit.loadIndices()")
smiles_batch = ["N#Cc1ccc(-n2nnc3ccccc32)cc1C#N",
                "COc1nn(C)cc1C(=O)NCC(OC)c1ccccc1C",
                "CCc1c(O)c(OC)c(Cc2cccnc2)c2nc(NC)sc12"]
#similarity search
res = g.query(f"UNWIND {repr(smiles_batch)} as smiles_batch \
RETURN rdk_sim_lsh(smiles_batch, .5) as result")
for query, result in zip(smiles_batch, res['result']):
    display(Draw.MolsToGridImage([Chem.MolFromSmiles(smiles) for smiles in res.iloc[0]['result'][:5]],
                                 subImgSize=(250,250), molsPerRow=5))
```

```
operation progress: 🟩 52/? [326.81op/s, done]
```





Speedup of katana vs postgres

Substructure index

- Distributed substructure index is sharded across hosts
- For each bit in the fingerprint, each host keeps a count of the number of molecules which set that bit in their pattern fingerprint, as well as a collection of their ids.
- The query molecule is compared to all molecules which share the least common bit with it (graph isomorphism)
- 7x faster than postgres on 1K random query molecules

Query molecule	1	0	1	0

Molecule 1	0	0	0	1
Molecule 2	1	0	1	1
Molecule 3	1	0	0	0
Bit position	0	1	2	3

Query molecule is compared to molecule 2 only

Example cypher query with heterogeneous data

Cypher:

```
MATCH (n: compound)-[:CID]-(:PubChem_BioAssay)-[]-(:gi)-[]-(:uniprot)-[]-(gene:gene)
```

```
WHERE n.canonical_smiles IN substructure_search("C(Nc1ccccc1)Nc1ccc(Oc2ccncc2)cc1")
```

```
RETURN gene.label, count(distinct n)
```

English:

For every compound-to-gene path

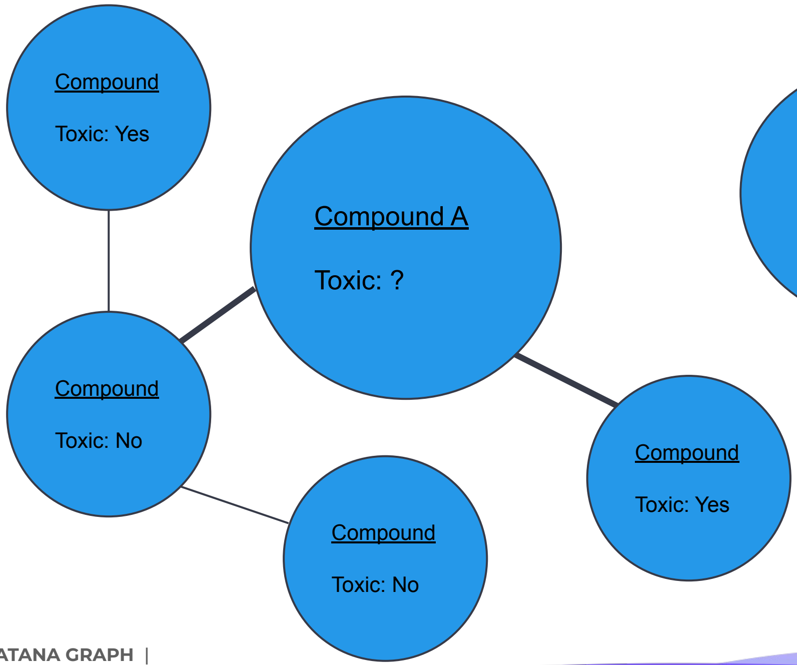
Where the compound contains a given substructure

Return the name of the gene and how many compounds with the given substructure interact with it

Beyond Querying

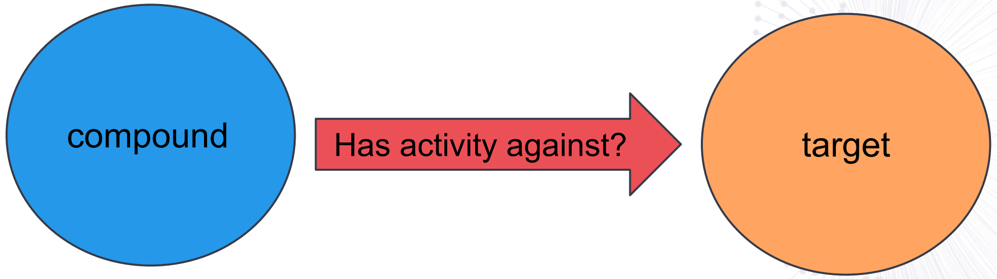
Node classification/regression

Predict values on nodes in graph



Link prediction

Predict graph topology



TDC ADMET Benchmark Challenge

Therapeutics Data Commons (TDC)

THERAPEUTICS DATA COMMONS

https://tdcommons.ai

HARVARD UNIVERSITY Georgia Tech Carnegie Mellon University Stanford University

Massachusetts Institute of Technology IQVIA ILLINOIS

- ADMET Group Challenge: **A**bsorption, **D**istribution, **M**etabolism, **E**xcretion, and **T**oxicity prediction tasks

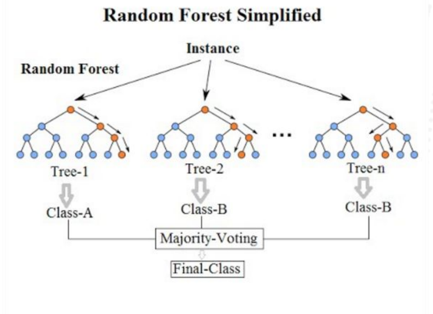
Given a drug candidate's structural information (SMILES), predict its ADMET profile.

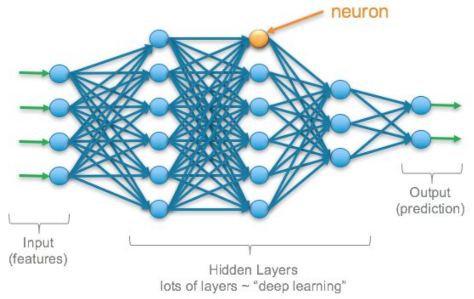
- 22 ADMET Datasets in TDC.
  - Binary Classification or Regression tasks
  - Fixed Evaluation Metric
  - Scaffold split for test data

hERG blockers	•Classification: predict whether a drug will block hERG or not.
Caco-2	•Regression: predict the Caco-2 cell effective permeability
BBB (Blood-Brain Barrier)	•Classification: predict whether a drug will penetrate blood-brain barrier.

Molecular Property Prediction: Current Approaches

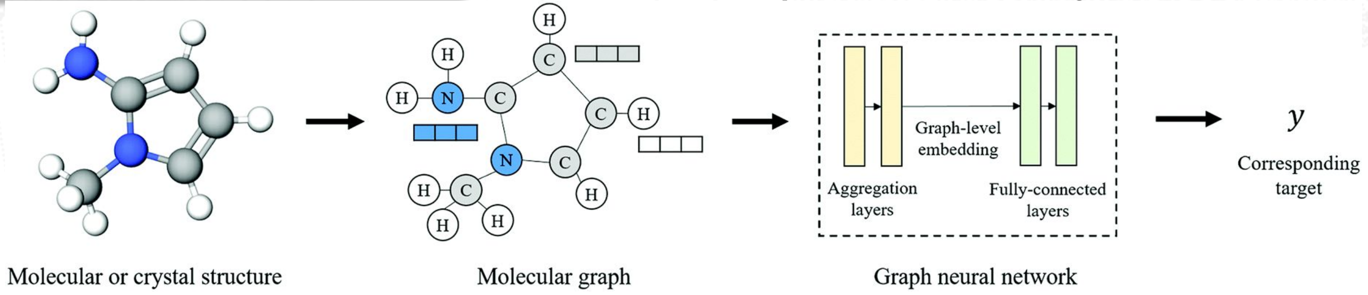
Fingerprint Based ML





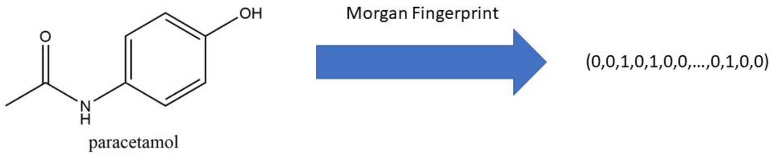
Graph Representation Based ML

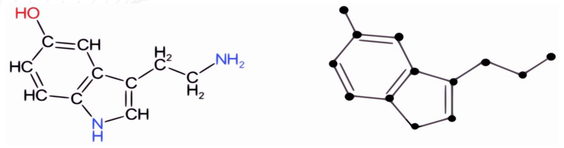
Graph Classification/Regression task.



Molecular Property Prediction: Current Approaches

- Fingerprint Based ML
  - Traditional Models
    - Random Forest
    - XGBoost
    - LighGBM
  - Deep Learning
    - MLP
- Graph Representation Based ML
  - Supervised Learning
    - GCN
    - GIN
    - AttentiveFP
  - Unsupervised Learning
    - Grover
    - ContextPred
    - AttrMasking





SimGCN: A New Approach towards Molecular Property Predictions

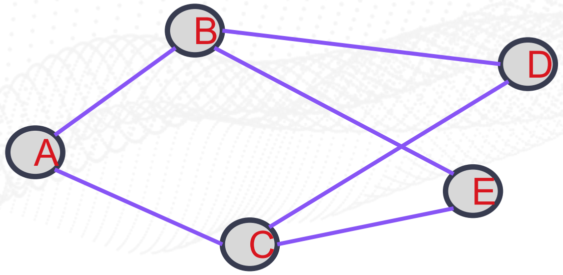
- How can we exploit the structural similarity of the molecules?

SimGCN

- Construct a **Similarity Graph** based on Tanimoto Similarity of the molecules
- Train a GNN (GCN, GAT) model on the similarity graph: **Node Classification/Regression task**.

Similarity Graph Construction

- Threshold Graph: link two drugs if their similarity > threshold
- KNN Graph: Connect each drug to its k most similar drugs

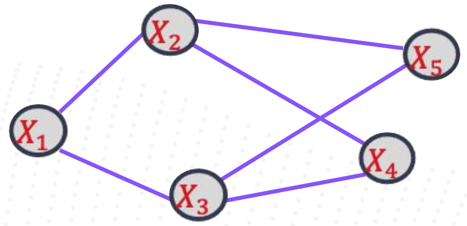


https://github.com/KatanaGraph/SimGCN-TDC

SimGNN Pipeline for Molecular Property Predictions

Node Classification Based Approach: Domain-Specific Knowledge Infusion

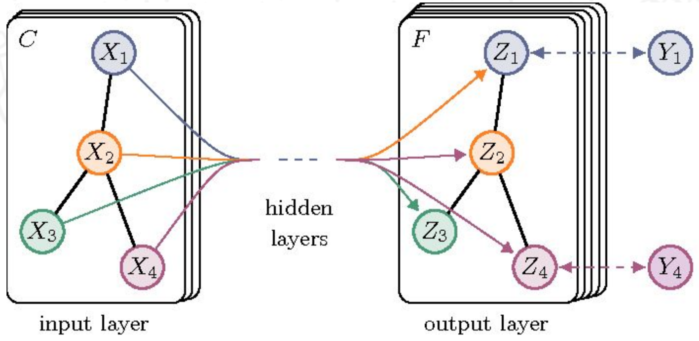
Drug	Y
CC(C)Nc1cccnc1N1CCN(C(=O)C2=CC3=C[C@H](NS(C)(=...	0.0
O=C1C=CC[C@@H]2[C@H]3CCCN4CCC[C@H](CN12)[C@H]34	0.0
O=C(c1ccc(OCC[NH+]2CCCCC2)cc1)c1c(-c2ccc(O)cc2...	0.0
Cc1cc2c(s1)Nc1ccccc1N=C2N1CCNCC1	1.0
CCN(CC)CCNC(=O)c1c(C)[nH]c(/C=C2\C(=O)Nc3ccc(F...	1.0

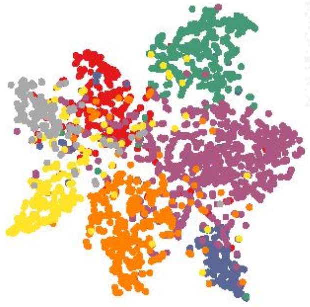


Katana RDkit Similarity

Node Features: Katana HLS module

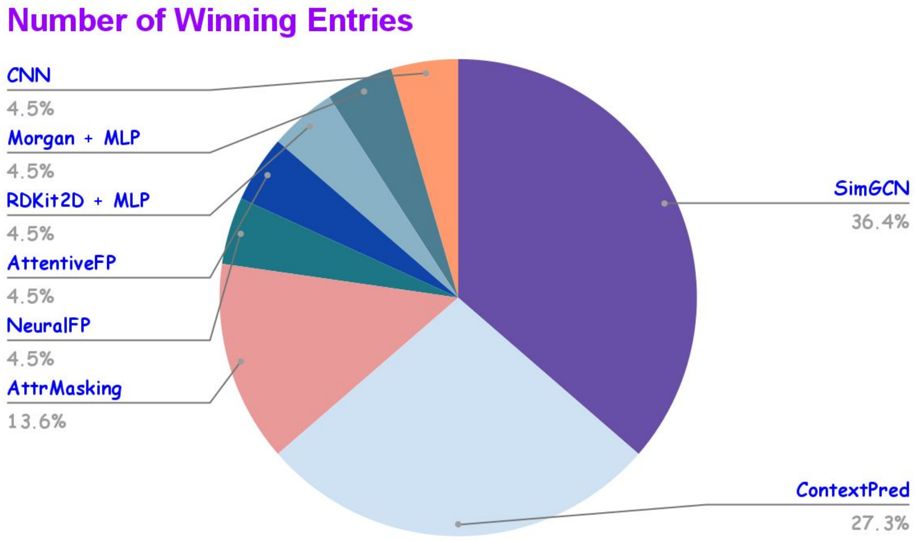
GNN

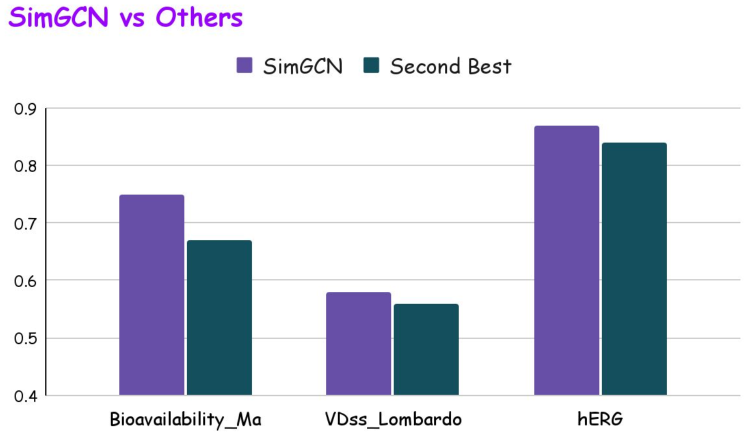




Success of SimGCN TDC ADMET Benchmarks

- SimGCN has the highest number of leading entries!



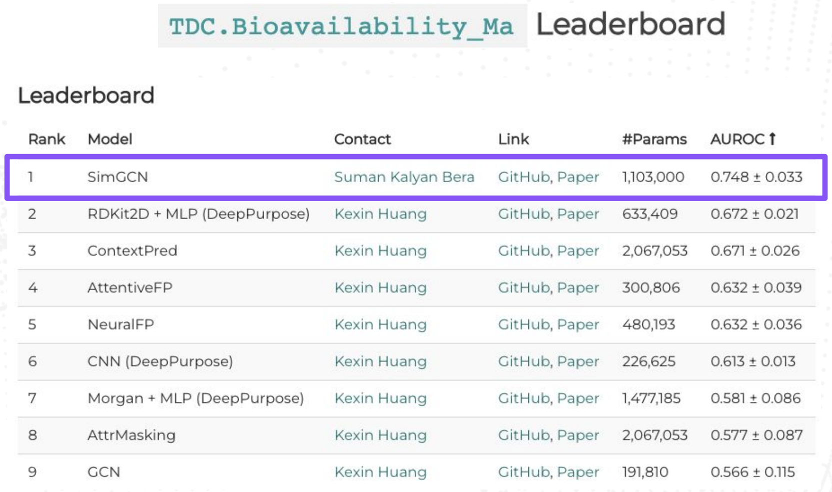


Tasks: absorption/distribution/metabolism/toxicity prediction

Datasets: `Half_Life_Obach, Clearance_Microsome_AZ, BBB_Martins, Pgp_Broccatelli, CYP2C9_Substrate_CarbonMangels etc.`

Success of SimGCN in TDC ADMET Benchmarks

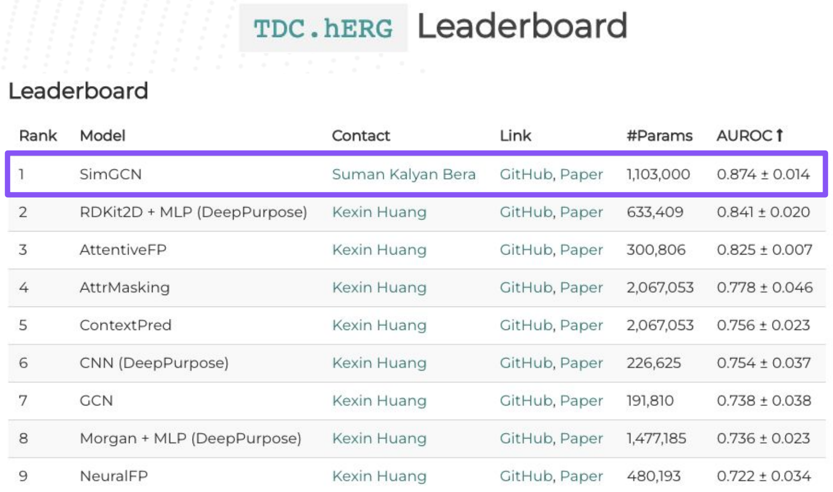
- Some of our entries at TDC leaderboard (https://tdcommons.ai/)



`TDC.Bioavailability_Ma` Leaderboard

Leaderboard

Rank	Model	Contact	Link	#Params	AUROC ↑
1	SimGCN	Suman Kalyan Bera	GitHub, Paper	1,103,000	0.748 ± 0.033
2	RDKit2D + MLP (DeepPurpose)	Kexin Huang	GitHub, Paper	633,409	0.672 ± 0.021
3	ContextPred	Kexin Huang	GitHub, Paper	2,067,053	0.671 ± 0.026
4	AttentiveFP	Kexin Huang	GitHub, Paper	300,806	0.632 ± 0.039
5	NeuralFP	Kexin Huang	GitHub, Paper	480,193	0.632 ± 0.036
6	CNN (DeepPurpose)	Kexin Huang	GitHub, Paper	226,625	0.613 ± 0.013
7	Morgan + MLP (DeepPurpose)	Kexin Huang	GitHub, Paper	1,477,185	0.581 ± 0.086
8	AttrMasking	Kexin Huang	GitHub, Paper	2,067,053	0.577 ± 0.087
9	GCN	Kexin Huang	GitHub, Paper	191,810	0.566 ± 0.115



`TDC.hERG` Leaderboard

Leaderboard

Rank	Model	Contact	Link	#Params	AUROC ↑
1	SimGCN	Suman Kalyan Bera	GitHub, Paper	1,103,000	0.874 ± 0.014
2	RDKit2D + MLP (DeepPurpose)	Kexin Huang	GitHub, Paper	633,409	0.841 ± 0.020
3	AttentiveFP	Kexin Huang	GitHub, Paper	300,806	0.825 ± 0.007
4	AttrMasking	Kexin Huang	GitHub, Paper	2,067,053	0.778 ± 0.046
5	ContextPred	Kexin Huang	GitHub, Paper	2,067,053	0.756 ± 0.023
6	CNN (DeepPurpose)	Kexin Huang	GitHub, Paper	226,625	0.754 ± 0.037
7	GCN	Kexin Huang	GitHub, Paper	191,810	0.738 ± 0.038
8	Morgan + MLP (DeepPurpose)	Kexin Huang	GitHub, Paper	1,477,185	0.736 ± 0.023
9	NeuralFP	Kexin Huang	GitHub, Paper	480,193	0.722 ± 0.034

Takeaways

- Chemical Data as graphs: Powerful unified representation
  - Unlocks new analytics and machine learning capabilities
  - Break data silos
- Distributed computation at scale required to search billion scale enumerated datasets.
- Unified platform for end-to-end pipelines
- Katana Graph is building the platform for doing this at scale