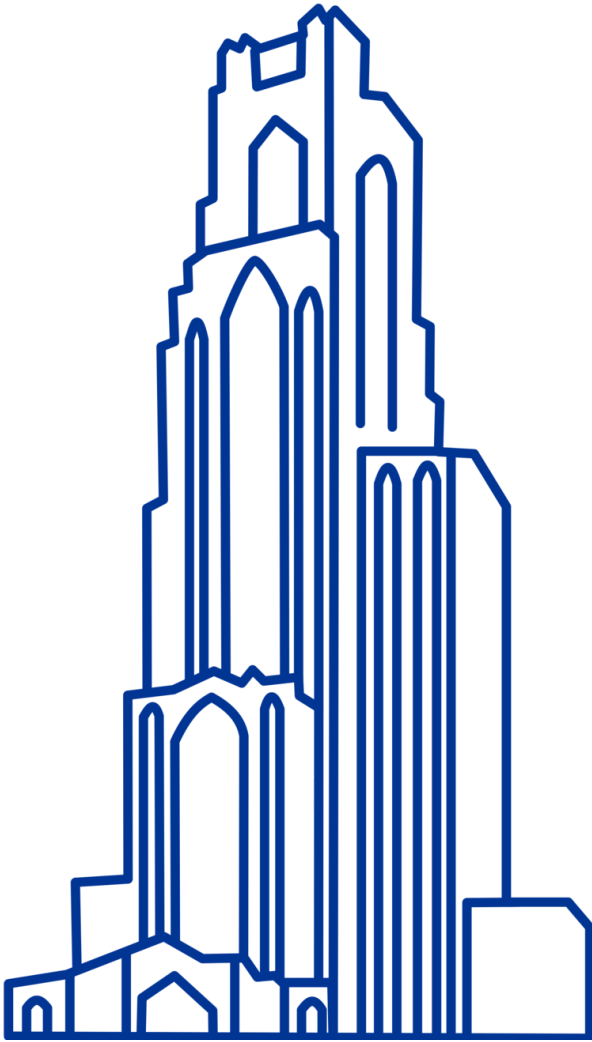


Computational Biology

(BIOSC 1540)

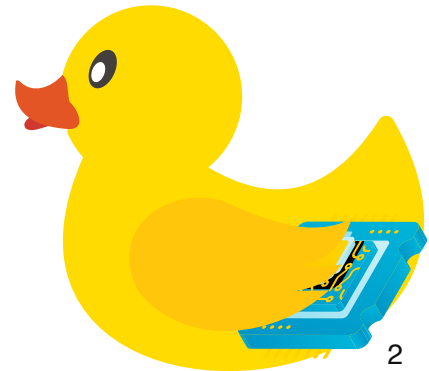
Lecture 18: Ligand-based drug design

Nov 7, 2024

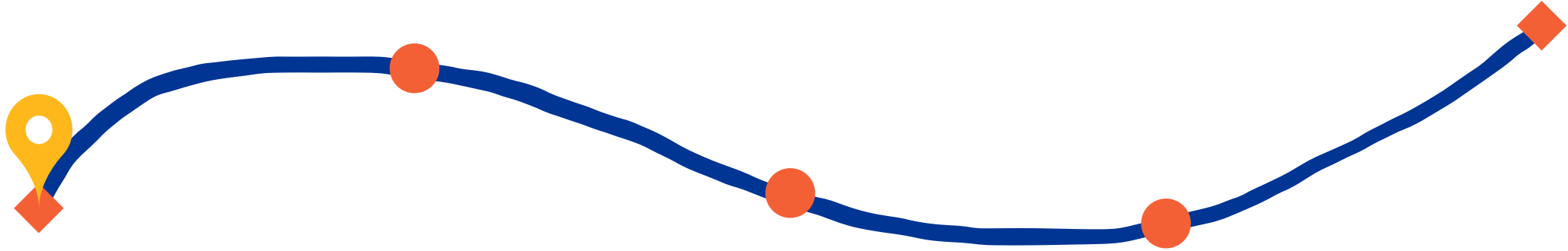


Announcements

- [A07](#) is due Thursday by 11:59 pm
- CSB exam is next Thursday (Nov 14)
 - Study guide will be posted tonight or tomorrow
 - We will have a review session on Tuesday (Nov 12)
 - Request DRS accommodations if needed
- [Project](#) will be due Dec 10
- OMETs will be coming out soon
- Attending our optional Python lectures are strongly recommended if you are taking simulation or modeling



After today, you should have a better understanding of



The basic principles of ligand-based drug design and how it differs from structure-based approaches.

Structural insight into a disease is a privilege

Phenotypic drug screening involves testing compounds on an organism level to identify potential leads

Example: Drug screening on an antibiotic-resistant bacterial strain to identify potential new leads



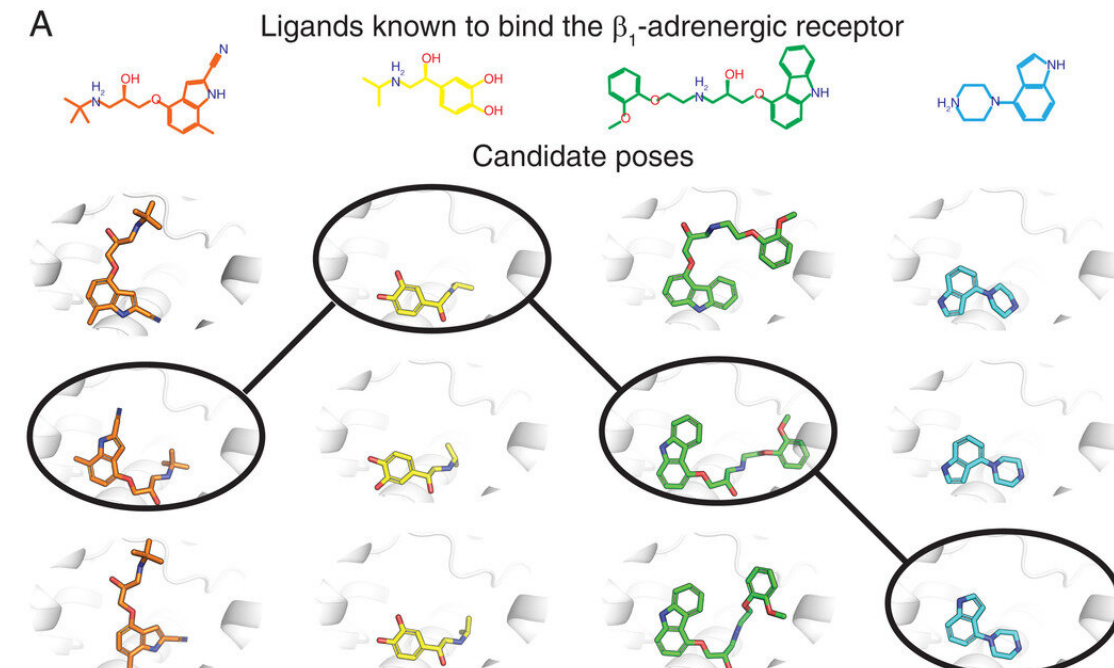
LBDD uses known compounds to guide drug discovery

Ligand-based drug design (LBDD) relies on the properties of known bioactive compounds

LBDD does not **require** the structure of the target protein, making it useful when this is unknown

Motivation: If we find compounds with little bioactivity, we can use LBDD to find compounds with similar chemical features to improve specific outcomes

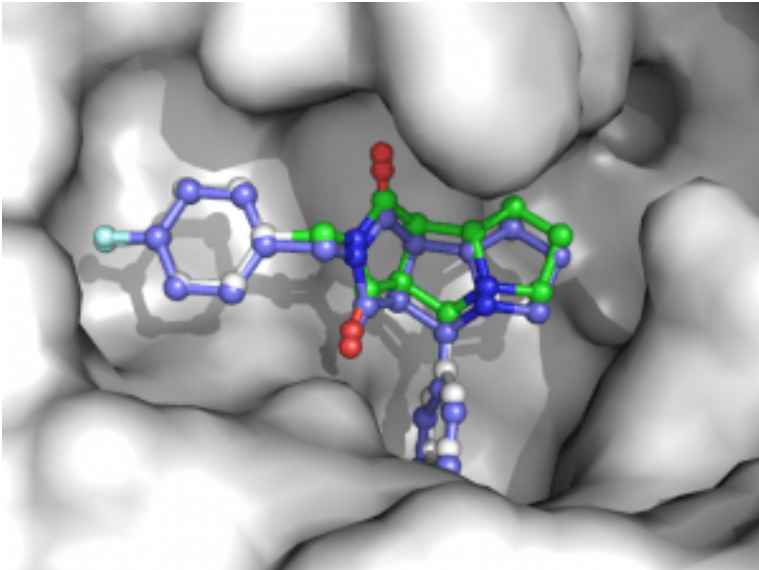
Assumption: Similar structures can lead to similar—hopefully improved—biological effects



Key differences between structure- and ligand-based drug design

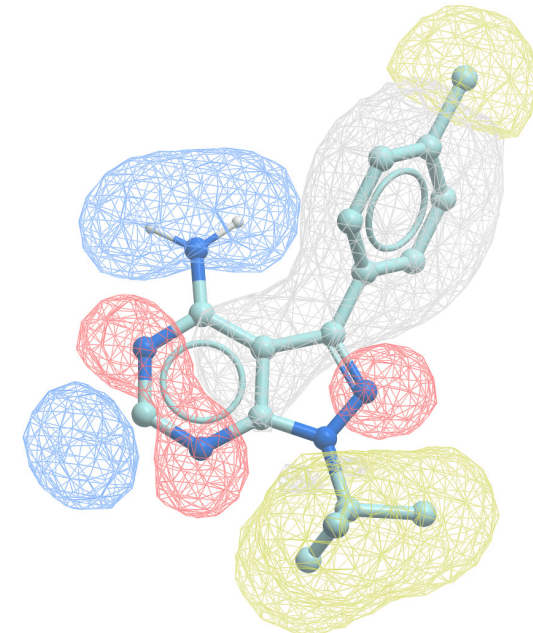
Structure-Based Drug Design:

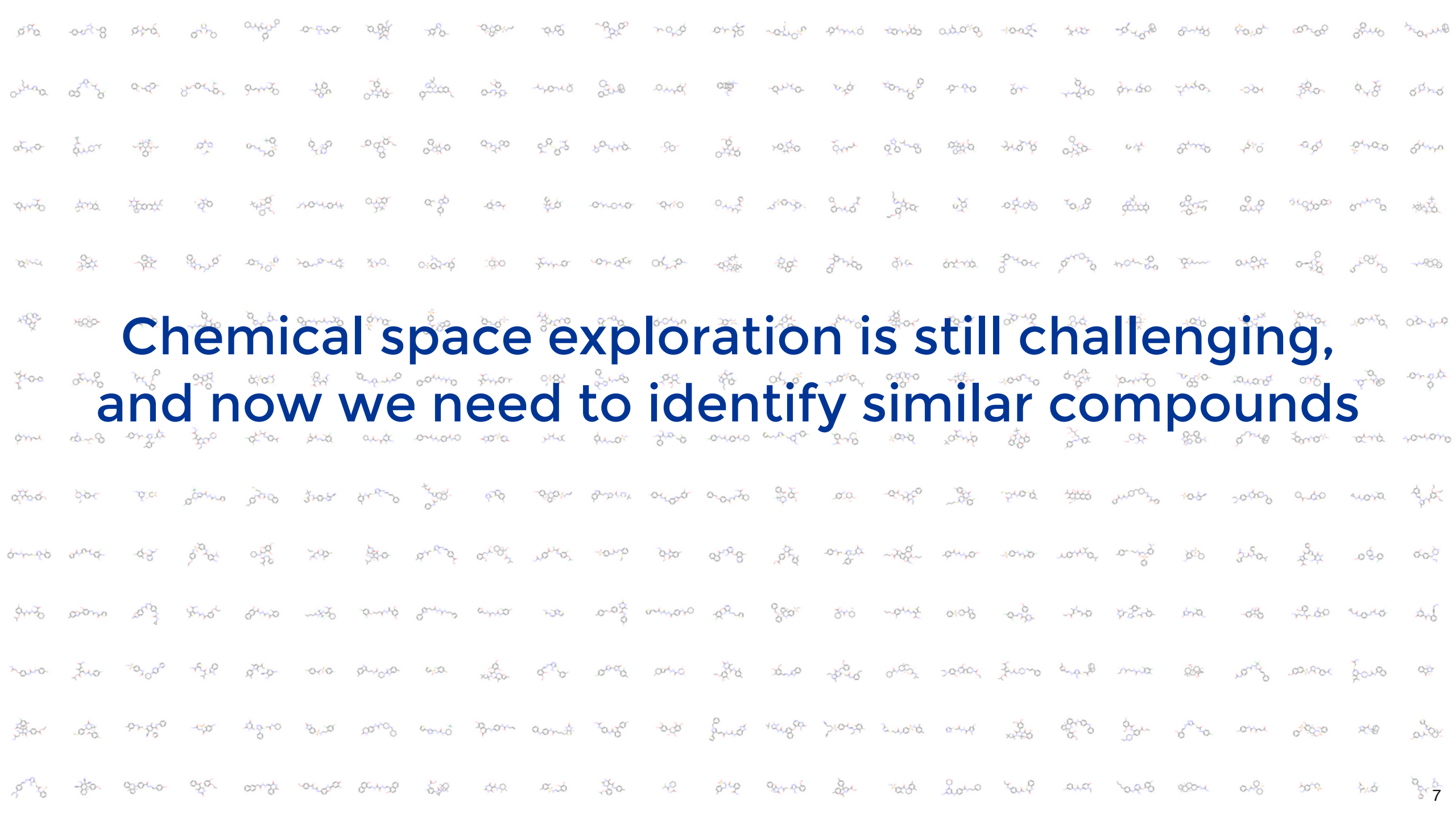
- Requires 3D structure of the target protein.
- Uses the binding site structure to model potential interactions.
- Often employs docking and molecular simulations.



Ligand-Based Drug Design:

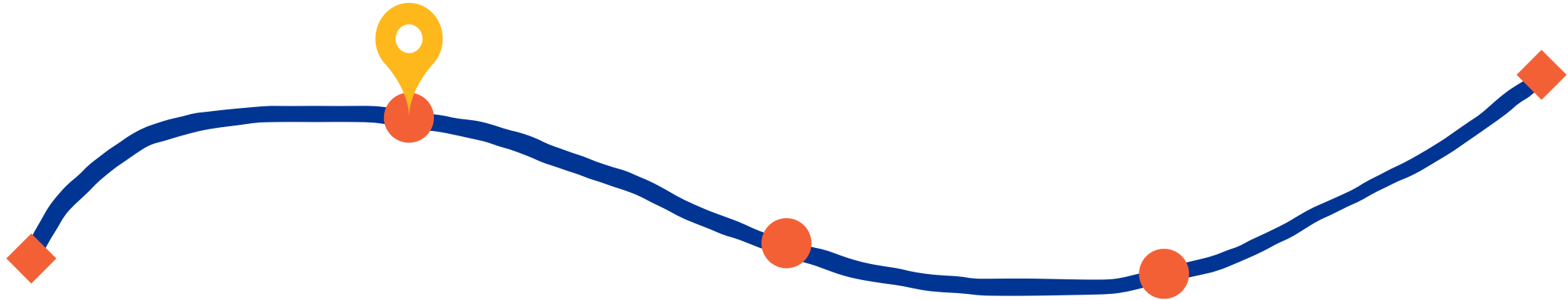
- Requires no structural information of the target.
- Uses the chemical structure and activity of known ligands as guides.
- Relies on molecular similarity rather than direct binding predictions.





**Chemical space exploration is still challenging,
and now we need to identify similar compounds**

After today, you should have a better understanding of

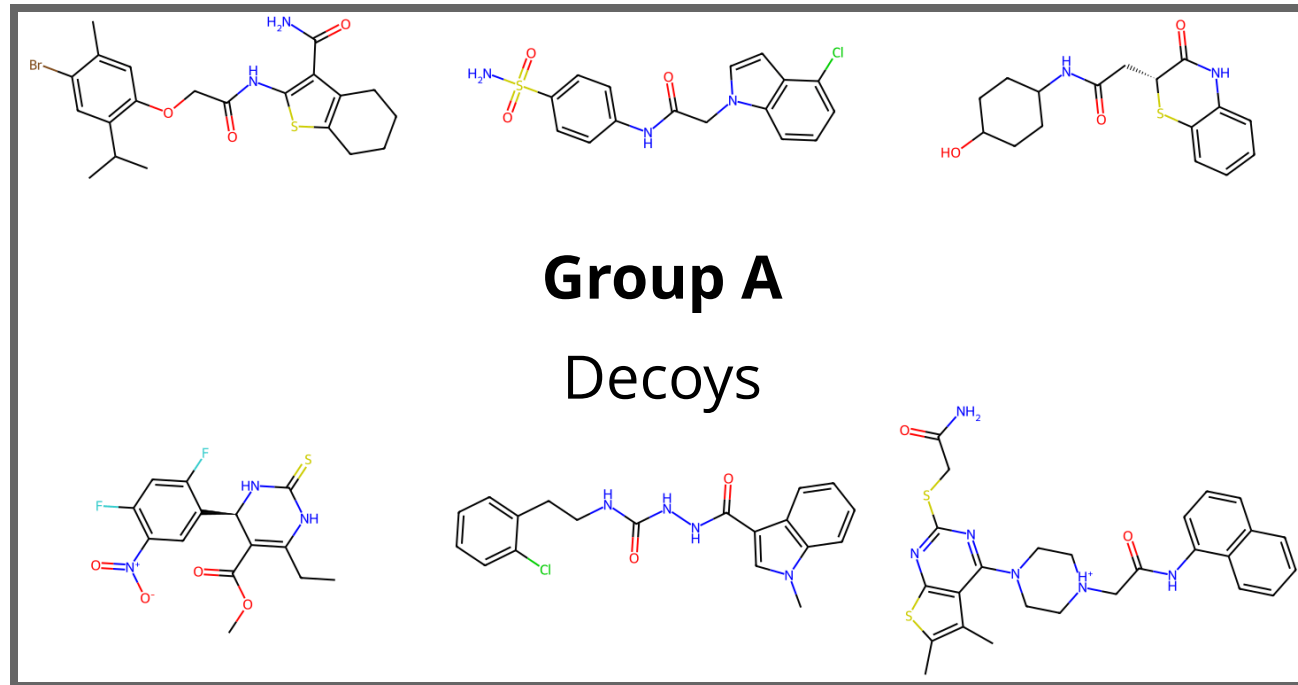
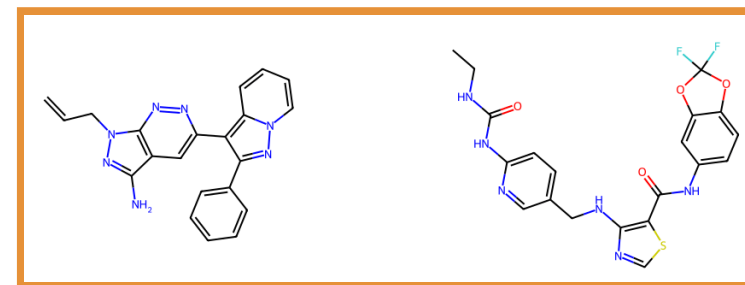


How descriptors and fingerprints evaluate
molecular similarity.

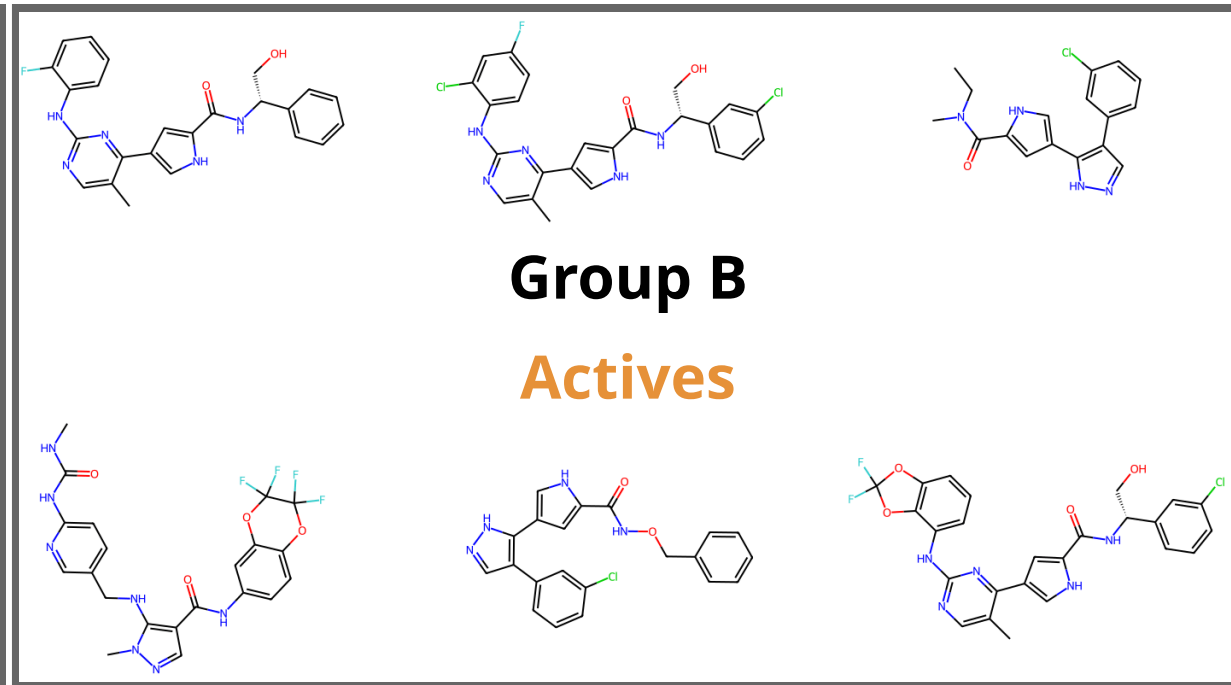
Quantifying molecular similarity is challenging

Suppose we performed an experimental high-throughput screen and identified these **potential leads**

Which group of molecules should we pursue for increased bioaffinity?



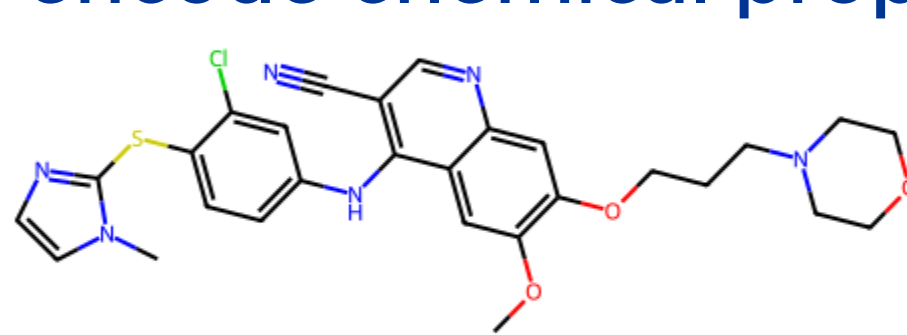
Group A
Decoys



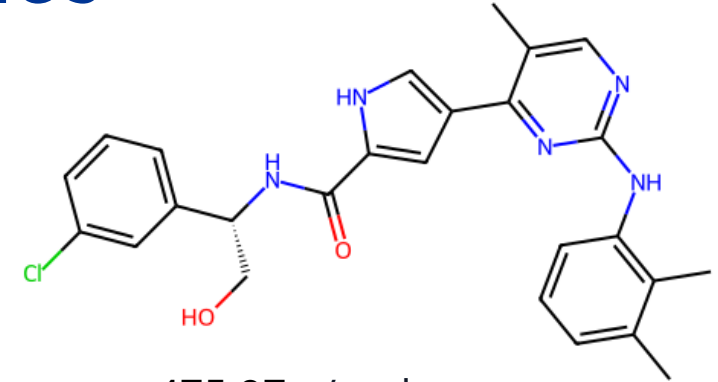
Group B
Actives

With your neighbors, determine how you would choose the group of molecules to pursue.

Molecular descriptors numerically encode chemical properties



565.09 g/mol



475.97 g/mol

Molecular weight

Indicates the overall size of the molecule, impacting drug distribution and elimination rates in the body.

LogP

4.08

4.30

Measures lipophilicity, which influences a molecule's ability to cross cell membranes and affects absorption and bioavailability.

Molar Refractivity

156.23

134.72

Relates to polarizability and electron cloud distribution, affecting intermolecular interactions and binding affinity.

TPSA

122.76 Å²

102.93 Å²

Estimates the molecule's ability to form hydrogen bonds, impacting solubility and permeability across biological membranes.

Num. rotatable bonds

10

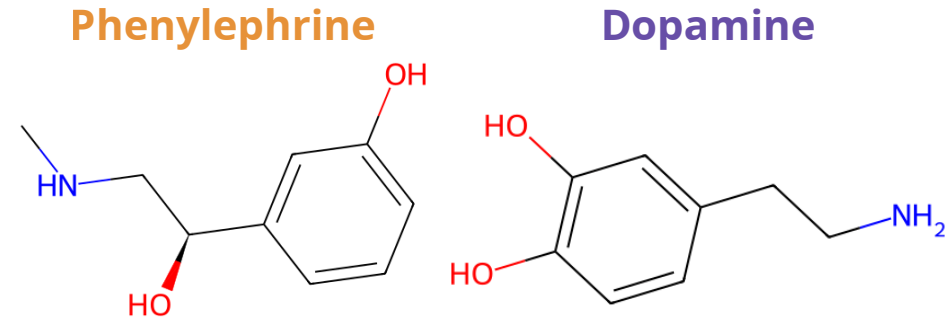
8

Reflects molecular flexibility, which can influence binding affinity and oral bioavailability.

Molecules can have similar properties, with slight structural differences causing widely different functions

Phenylephrine is a synthetic compound that acts as a vasoconstrictor by stimulating alpha-adrenergic receptors

Dopamine is a naturally occurring neurotransmitter in the brain and interacts with dopamine receptors



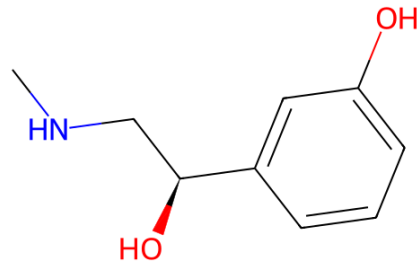
Molecular weight	167.21 g/mol	153.18 g/mol
LogP	0.65	0.46
Molar Refractivity	47.01	42.97
TPSA	52.49 Å ²	66.48 Å ²
Num. rotatable bonds	3	2
SMILES	<chem>CNC[C@H](C1=CC(=CC=C1)O)O</chem>	<chem>C1=CC(=C(C=C1CCN)O)O</chem>

Simple descriptor comparisons are not sufficient for computing molecular similarity

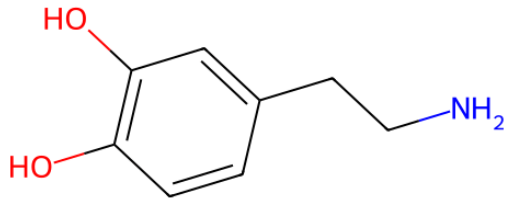
Molecular fingerprints encode structural information

Extended Connectivity Fingerprints (ECFPs) encode structural features into numerical representations

Phenylephrine



Dopamine

[illegible][illegible]

```
1 from rdkit import Chem
2 from rdkit.Chem import rdFingerprintGenerator
3 fmgen = rdFingerprintGenerator.GetMorganGenerator(
4     radius=3, fpSize=1024,
5     atomInvariantsGenerator=rdFingerprintGenerator.GetMorganFeatureAtomInvGen()
6 )
7 mol = Chem.MolFromSmiles("C1=CC(=C(C=C1CCN)O)O")
8 print(fmgen.GetFingerprint(mol))
```

How do we compute this?

Hash functions are used to encode chemical information

"Encoding" is a computational term for transforming information in a numerical format for computers

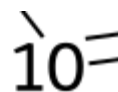
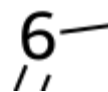
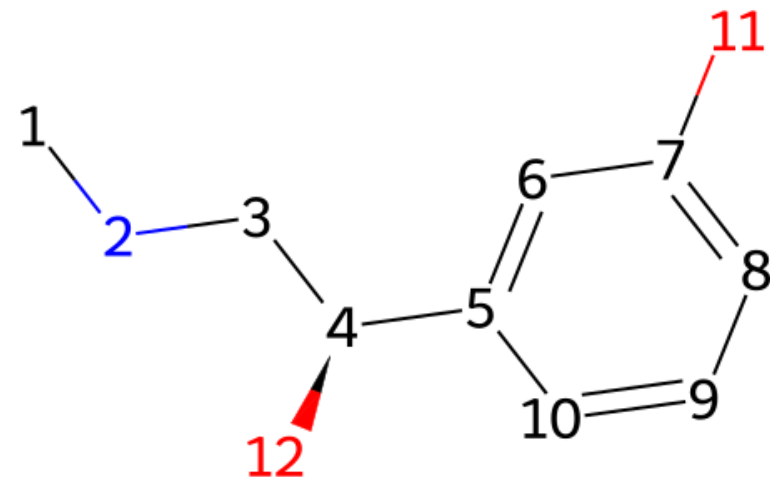
For each heavy atom (i.e., not H), hash atom-specific properties

$$ID_0 = \text{hash}(Z_i, V_i, C_i, R_i, \dots)$$

ID_0	Z	Atomic number
Iteration 0	V	Valence
identifier	C	Formal charge
	R	Ring membership

Let's look at carbons 6 and 10

Because of the same element and connectivity, they have the same ID_0



```
id6_iter0 = hash((6, 3, 0, 1))  
print(id6_iter0) # 7468469475583712974
```

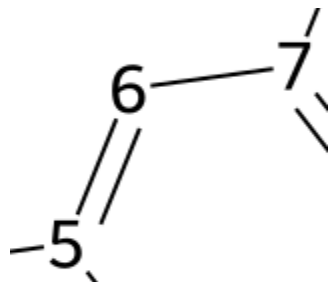


```
id10_iter0 = hash((6, 3, 0, 1))  
print(id10_iter0) # 7468469475583712974
```

For each additional iteration of n , incorporate the hashes of connected atoms that are n bonds away

Next, encode the atom IDs that are exactly one bond away

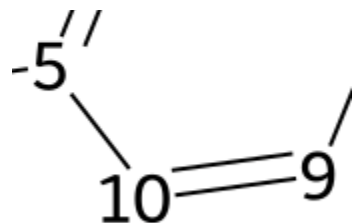
Format: (IterationNumber, AtomID, BondOrder1, AtomID1, BondOrder2, AtomID2, ...)



```
id6_iter1 = hash((
    1, 7468469475583712974, # ID for atom 6
    2, 901285887933171736, # ID for atom 5
    1, 901285887933171736 # ID for atom 7
))
print(id6_iter1) # -1070477880882296059
```

Repeat for all atoms while hashing $n - 1$ IDs

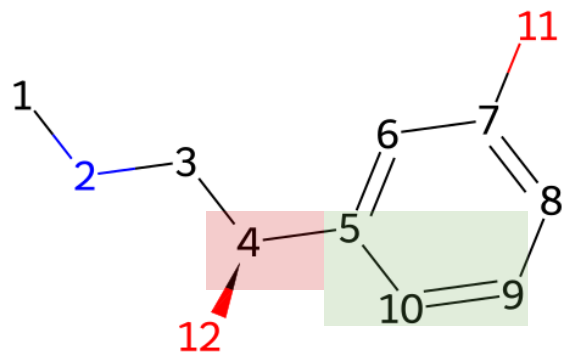
Each iteration encodes local chemical information into each atom's ID



```
id10_iter1 = hash((
    1, 7468469475583712974, # ID for atom 10
    1, 901285887933171736, # ID for atom 5
    2, 7468469475583712974 # ID for atom 9
))
print(id10_iter1) # 9113858623660175530
```

We can repeat the process for larger n , which captures more chemical information at a (small) computational cost

We keep track of atom IDs at each iteration to encode multiple "levels" of chemical information

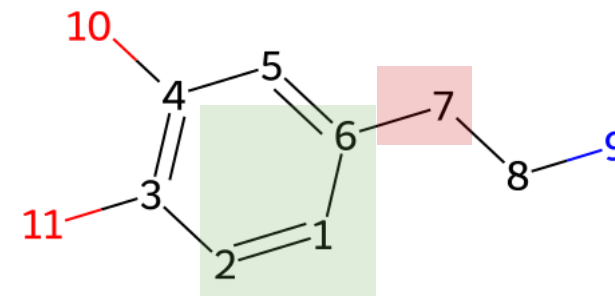


```
# Iteration 0
[-96873481, -5237400, -608624, -40896092, 13106358, 39304191,
13106358, 39304191, 39304191, 39304191, 18495798, 18495798]

# Iteration 1
[-12887828, 34836456, -82428984, -76182021, 57441373, 18535308,
36698099, -16062189, -71082609, -16062189, -13803757, -35226747]

# Iteration 2
[-30242937, -22342045, -3701095, -83323106, -81401022, -79585126,
259777, -18164777, -83853893, -9624634, -63890015, -86218719]

# Iteration 3
[24482285, -67056973, -1049934, 58183281, 9686245, 65319696,
-89546467, 90525418, -96278682, -31838946, -41820336, -42202112]
```



```
# Iteration 0
[39304191, 39304191, 13106358, 13106358, 39304191, 13106358,
-608624, -608624, -2248911, 18495798, 18495798]

# Iteration 1
[-16062189, -16062189, -54942758, -54942758, 18535308, 80518135,
-46276084, 85303560, -4225841, -13803757, -13803757]

# Iteration 2
[45202524, -32527659, 91315393, -86313403, 74663225, 43056615,
-92441264, 61456743, 35268850, -86729888, -86729888]

# Iteration 3
[17051553, -83857497, -10864101, 42020134, 84228020, 88509243,
53634925, 58427327, 85169475, -62345869, -23012595]
```

Similar structural features will share atom IDs
until our iteration starts incorporating different structural features

Atom IDs are encoded into a bit array

We can get a collection of atom IDs, but how would we rapidly compare molecules with different number of atoms?

We use **bit arrays**, which are fixed-length collections of ones and zeros

10101100

11011010

```
      10101100
AND   11011010
-----
      10001000
```

Features that are in **both molecules**

This allows efficient operations

```
      10101100
OR    11011010
-----
      11111110
```

Features that are in **either molecules**

Converting atom IDs to bit arrays

Decide on length of bit array, for example, 1024 and fill with zeros

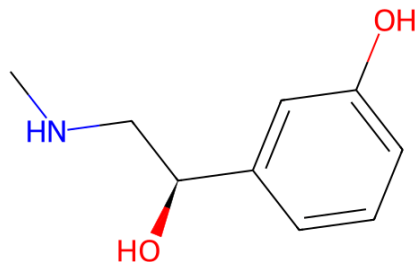
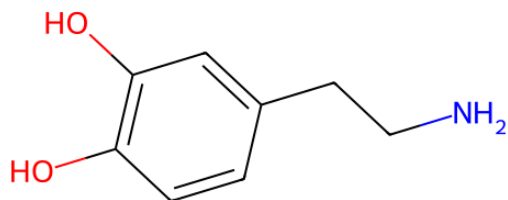
$$\text{ecfp} = [0, 0, 0, 0, \dots, 0, 0, 0]$$

Divide each atom ID by the length of the array and determine the remainder

$$-1070477880882296059 \bmod 1024 = 908$$

Set the value of the bit array at that index to 1

```
ecfp[908] = 1
```

[illegible][illegible]

Tanimoto similarity compares the ECFPs between two molecules

Molecular similarity: The concept that similar molecules often show similar biological effects.

Using bit operations, we can compute similarity using Tanimoto

$$\text{Tanimoto similarity} = \frac{c}{a + b - c}$$



```
a = len(fp1_bits)
b = len(fp2_bits)
c = len(fp1_bits & fp2_bits)
```

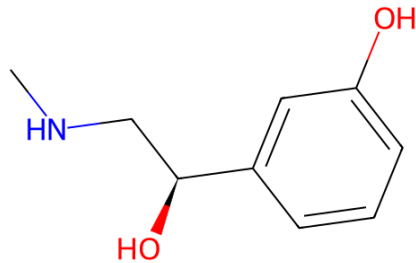
- a is the number of bits set to 1 in vector **A**.
- b is the number of bits set to 1 in vector **B**.
- c is the number of bits set to 1 in both vectors **A** and **B** (the intersection).

This formula measures the ratio of the shared features to the total number of unique features between two molecules.

Tanimoto similarity ranges

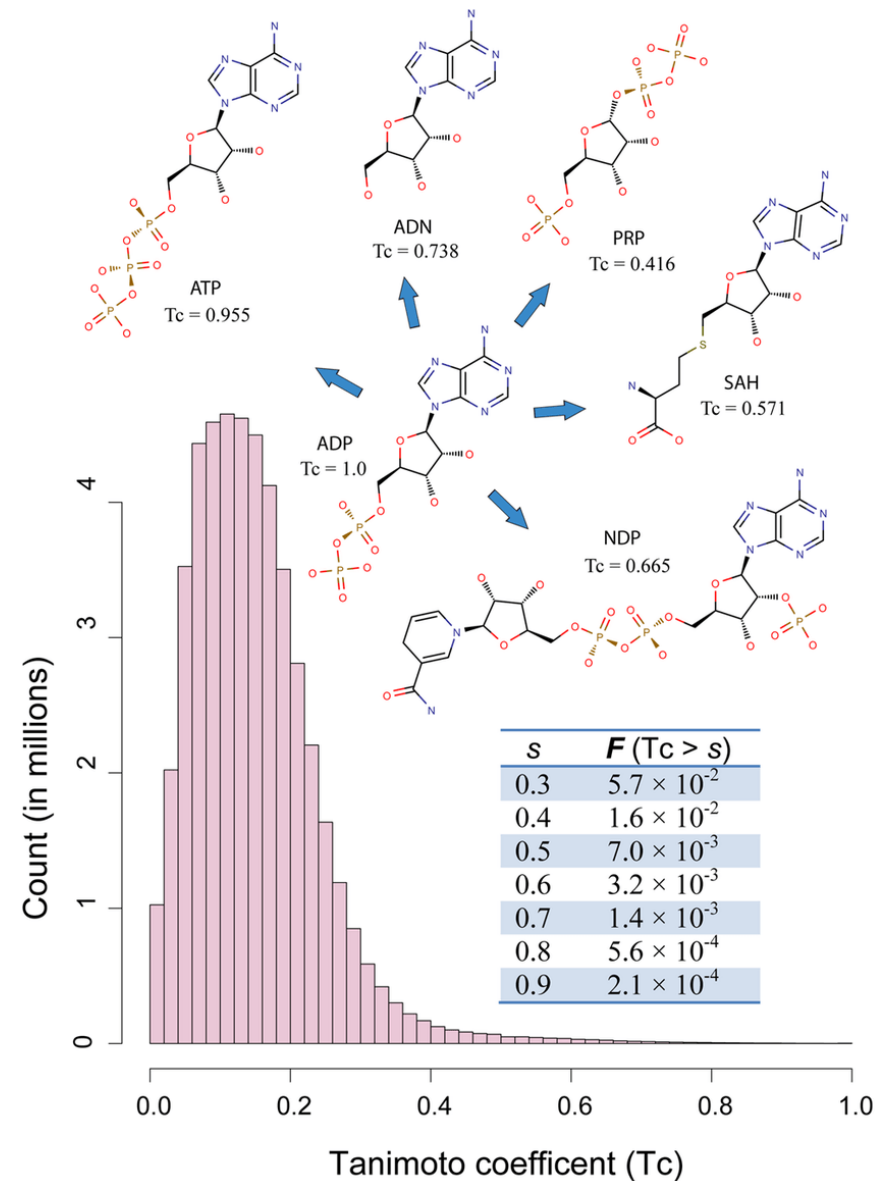
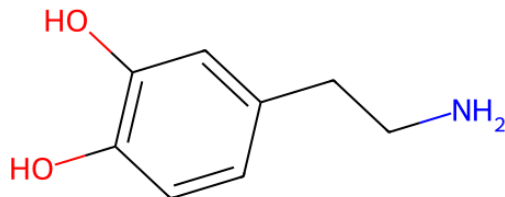
How similar does ECFPs and Tanimoto say these molecules are?

Phenylephrine



33%

Dopamine



After today, you should have a better understanding of



How QSAR models predict biological activity
based on molecular structure.

QSAR models link chemical structure with biological activity

Purpose: To predict the biological activity of molecules based on their structure.

Motivation:

- Reduces the need for experimental screening.
- Helps identify potential drugs quickly and cost-effectively.

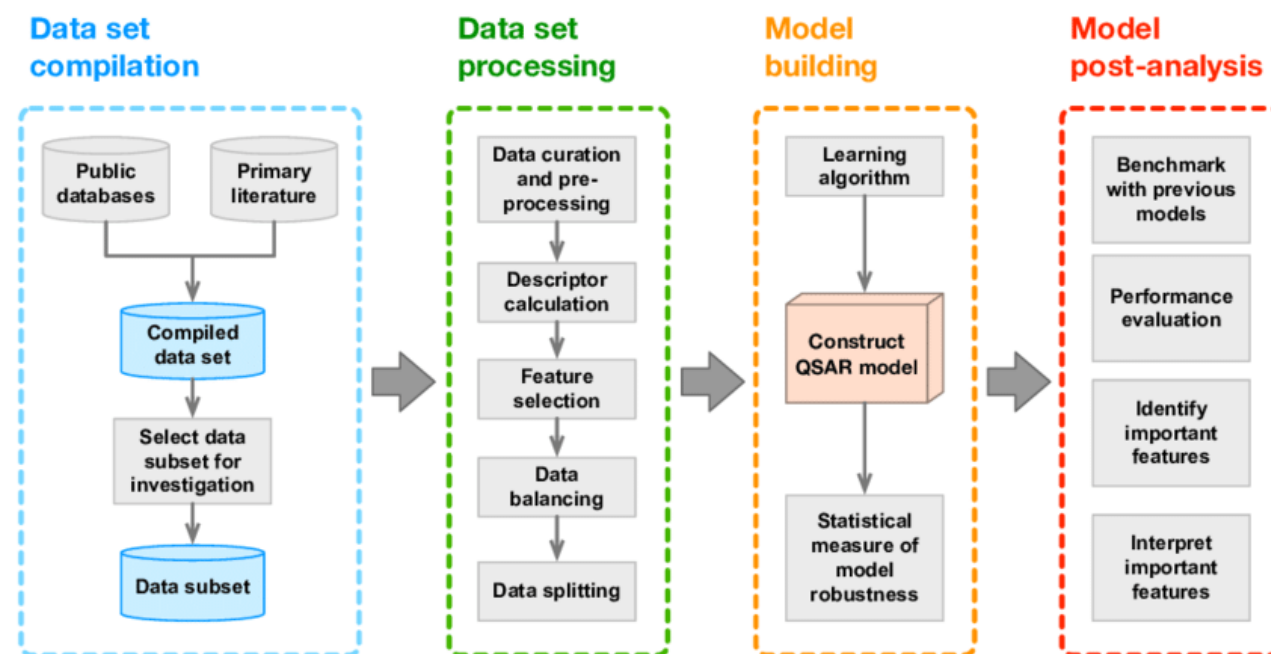
Example: Predicting if a compound is likely to be an inhibitor of a target enzyme based on known inhibitors.

Types of QSAR Models:

1. **Linear Models:** Simple, interpretable, e.g., linear regression.
2. **Nonlinear Models:** Capture complex relationships, e.g., neural networks.

Developing a QSAR model follows systematic steps

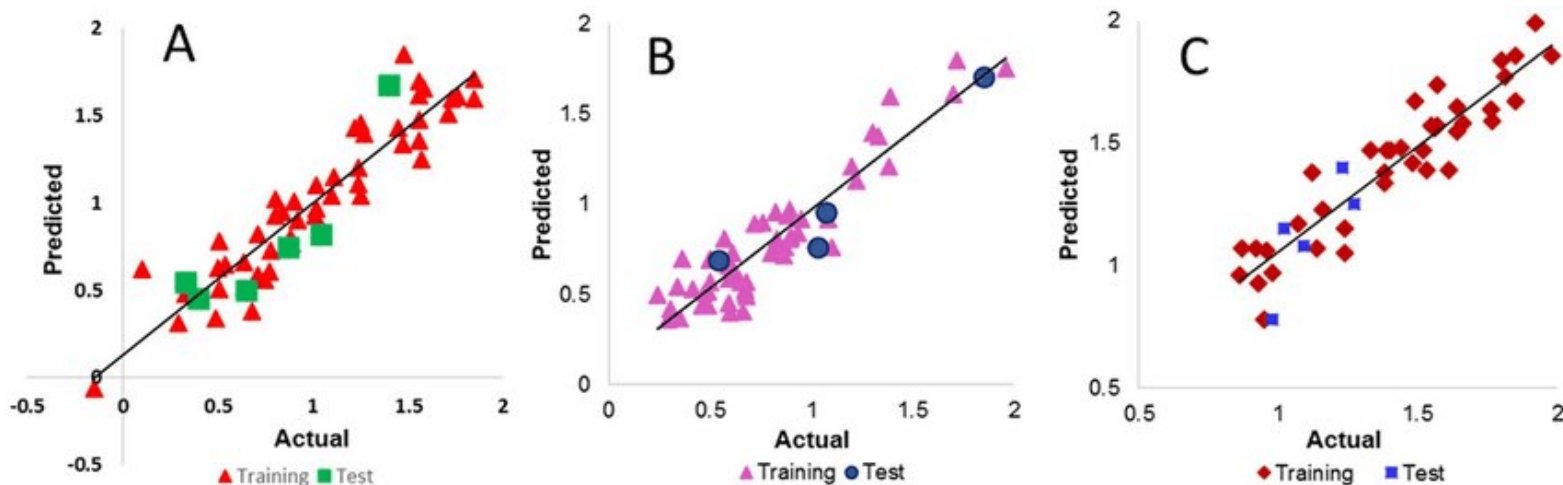
- **Data Collection:** Gather biological activity and molecular data.
- **Descriptor Calculation:** Calculate numerical descriptors for each molecule.
- **Model Selection and Training:** Use machine learning to correlate descriptors with activity.
- **Model Validation:** Test model accuracy with independent datasets.
- **Interpretation and Application:** Use the model for predicting new molecules.



Linear regression models are simple but effective for QSAR analysis

Fits a linear relationship between descriptors and output

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots$$

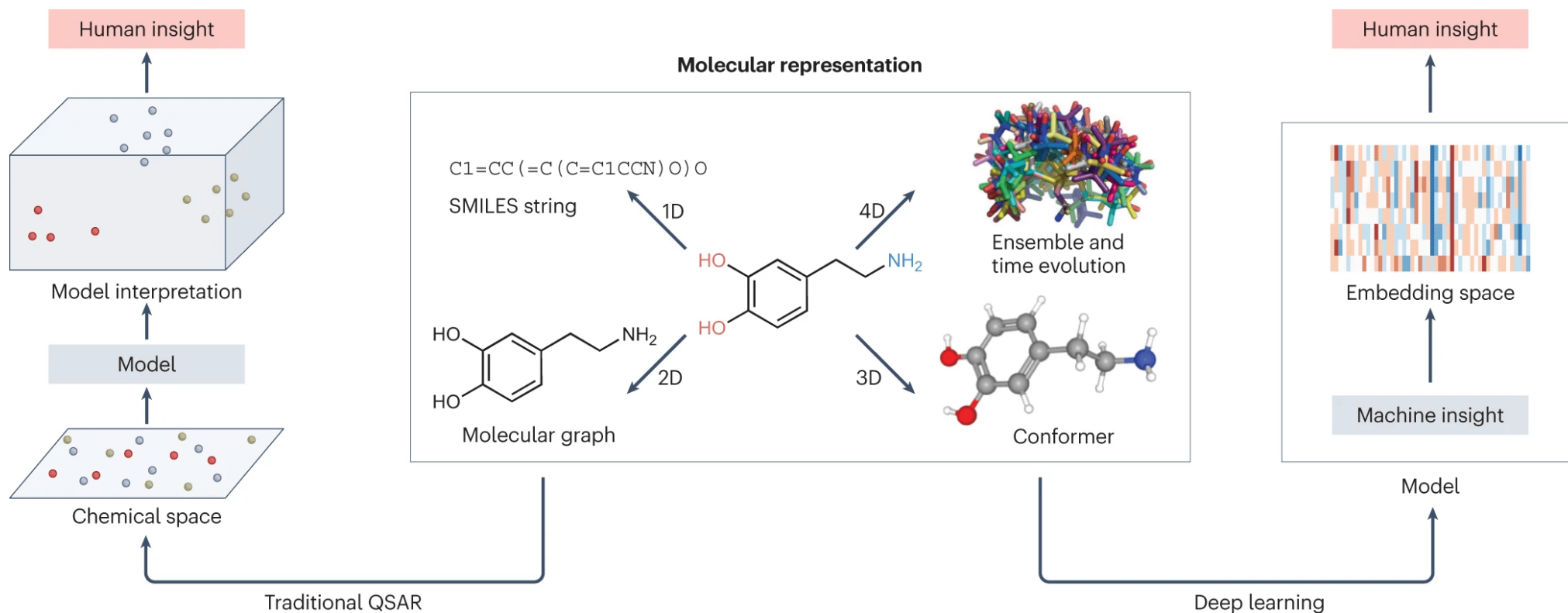


- **Advantages:** Easy to interpret.
- **Limitations:** Limited to linear relationships; struggles with complex datasets.

Nonlinear models capture complex relationships in QSAR data

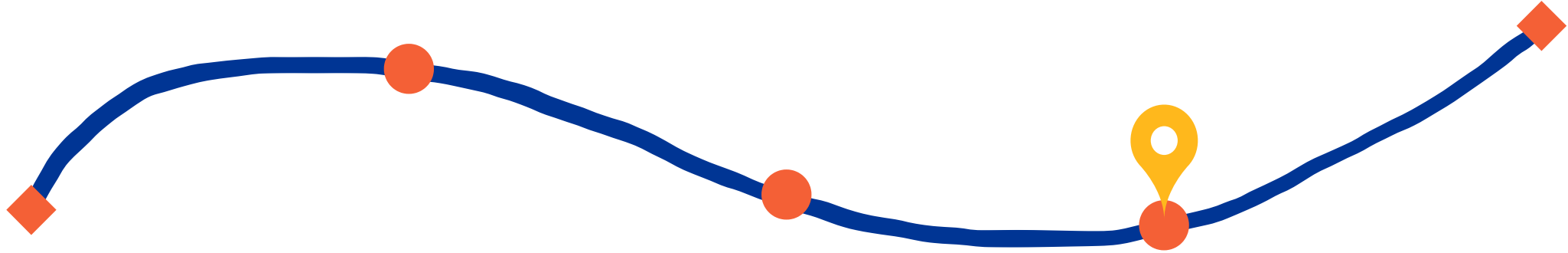
Examples of Nonlinear Models:

- **Neural Networks:** Capture complex, nonlinear patterns in large datasets.
- **Random Forests:** Effective for high-dimensional data, robust against overfitting.



Example: Predicting toxicity, where relationships between descriptors and outcomes are often nonlinear.

After today, you should have a better understanding of

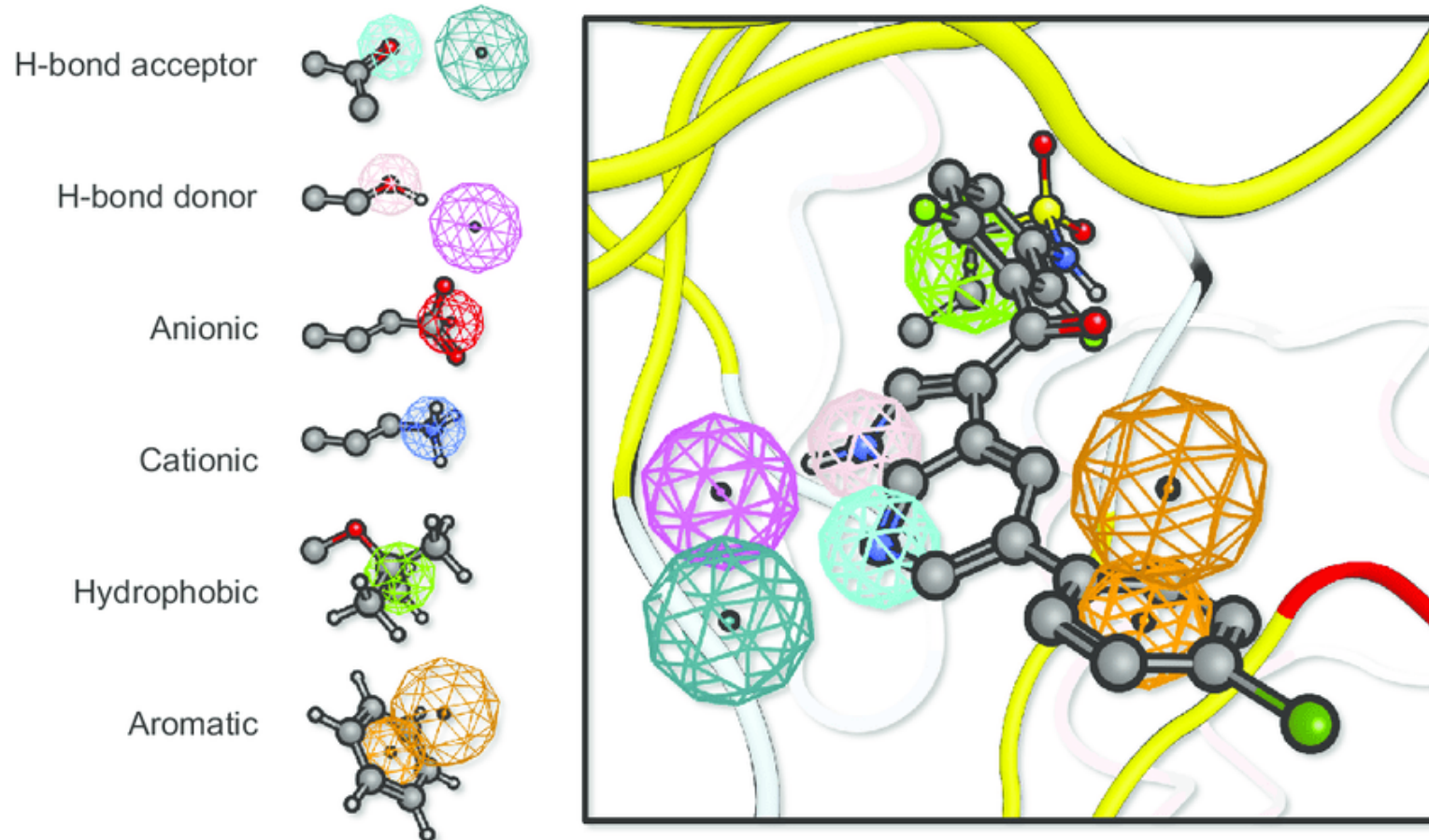


The role of pharmacophore modeling in identifying essential molecular features for activity.

**Where QSAR quantifies activity,
pharmacophore modeling identifies critical
molecular features for activity**

Pharmacophore modeling defines the essential features needed for biological activity

A pharmacophore is the 3D arrangement of molecular features required for biological activity



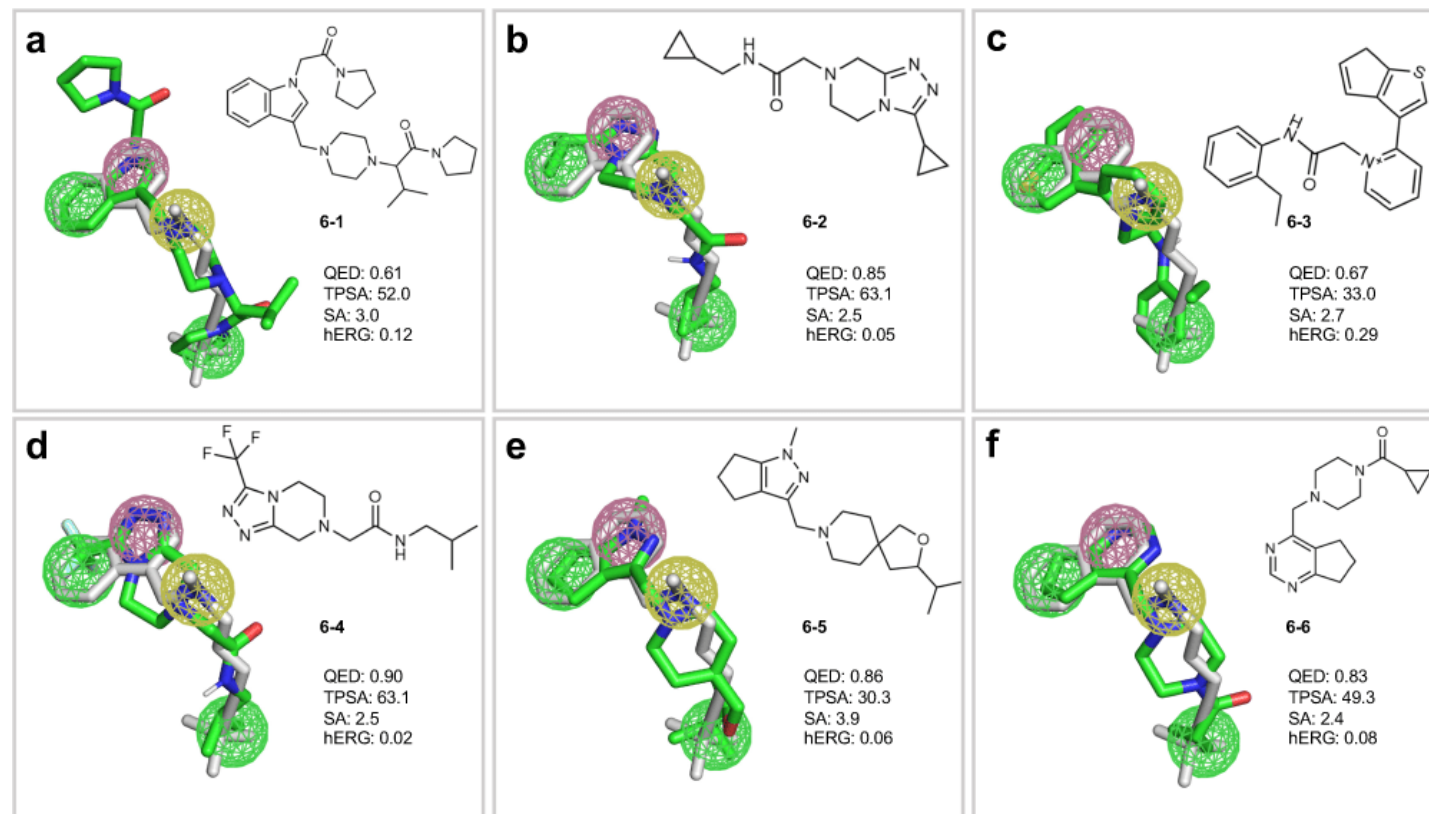
Building a pharmacophore model requires multiple active compounds

Step 1: Align active molecules

- Identify common structural features
- Determine spatial relationships
- Consider multiple conformations

Step 2: Define feature locations

- Mark shared pharmacophoric points
- Establish distance constraints
- Set tolerance spheres



Before the next class, you should

Lecture 18:

Ligand-based drug design

Exam 02 Review



Today



Tuesday

- Finish [A07](#)
- Study for exam