# EECS 6895 Advanced Big Data and AI

# Lecture 9: AI for Life Sciences III (Drugs)

Prof. Ching-Yung Lin

Columbia University

March 25th, 2025



COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# AI in BioMedical

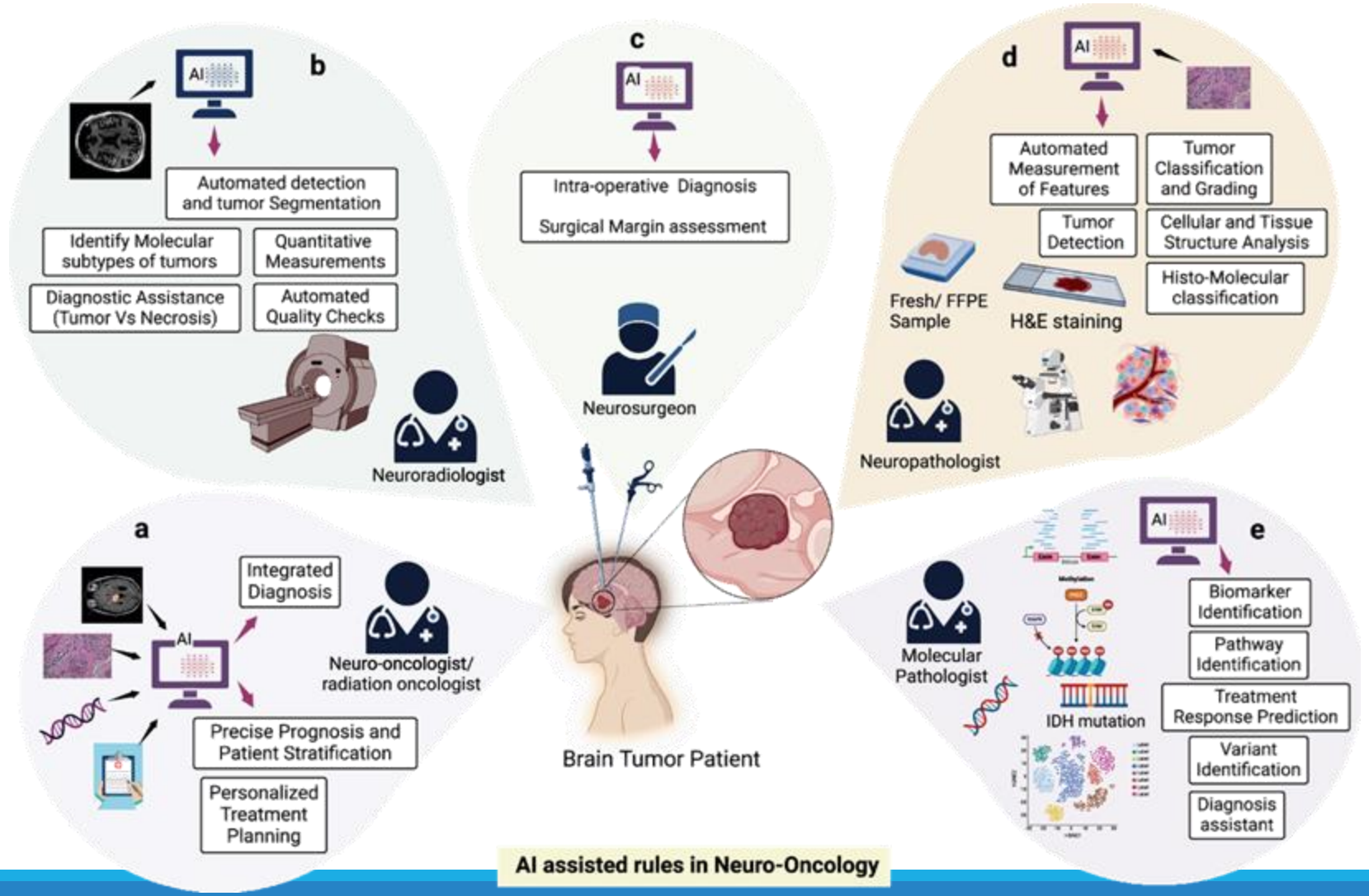
a) Through imaging, slicing, and blood-related patient information**, AI interprets the lesion and prognosis**

b) Image quantization and **segmentation**

c) Surgical guidance

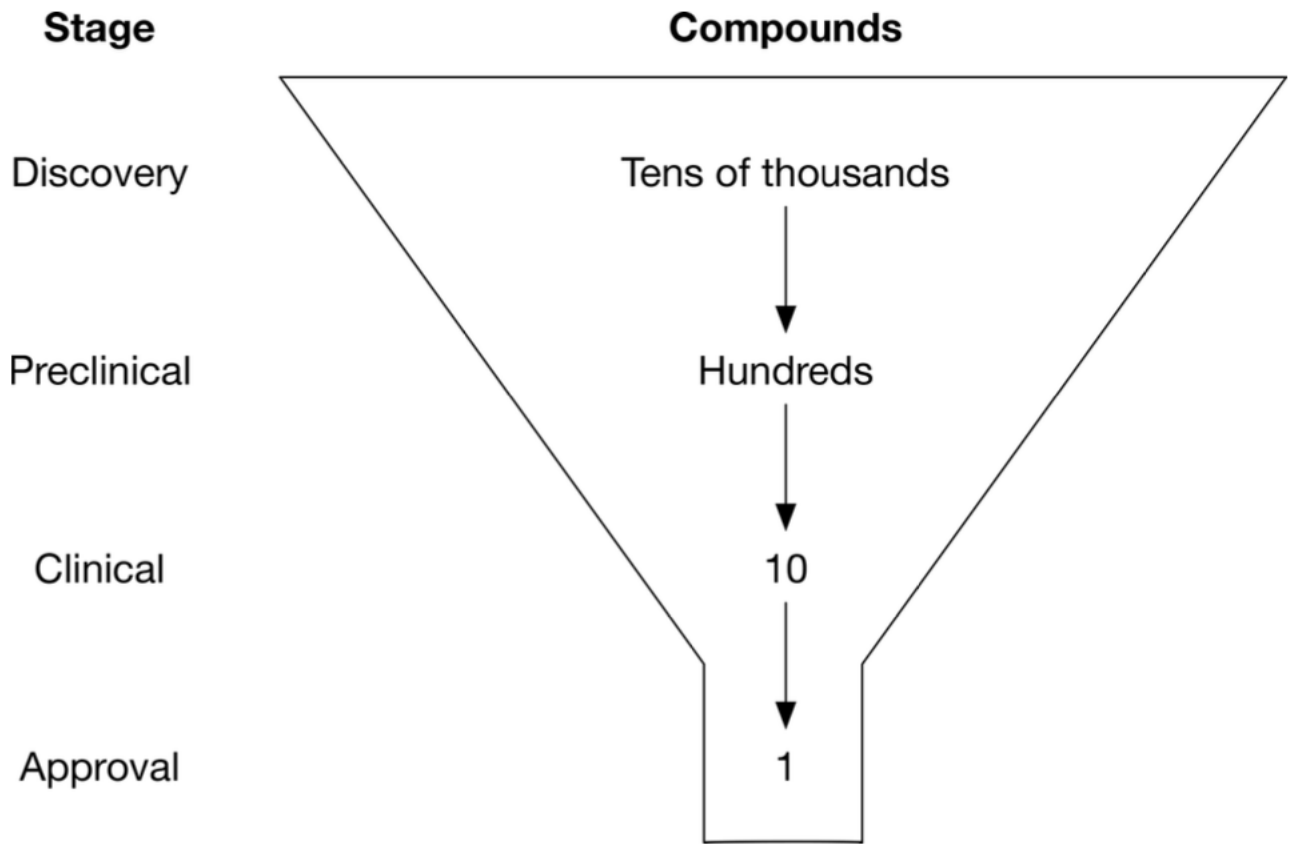
d) Disease clustering

e) Optimization of **drug strategies**, **finding druggable target**



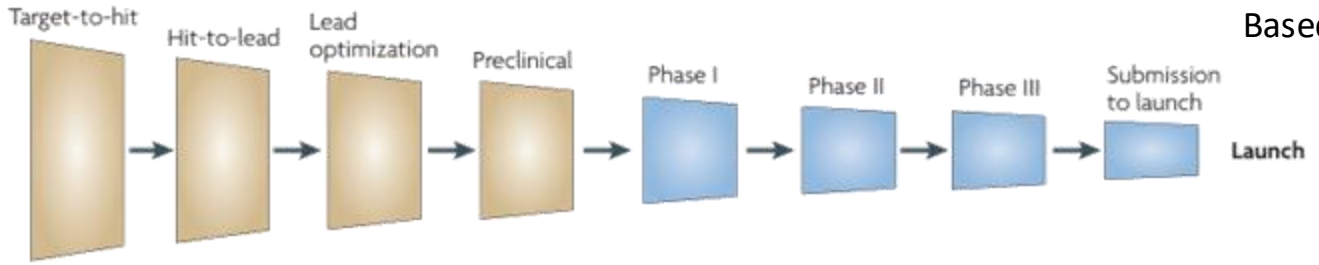

Khalighi, S., Reddy, K., Midya, A. et al. Artificial intelligence in neuro-oncology: advances and challenges in brain tumor diagnosis, prognosis, and precision treatment. npj Precis. Onc. 8, 80 (2024). https://doi.org/10.1038/s41698-024-00575-0

# Traditional Drug Development Funnel



# Steps of Drug Development -- Graphen 27x faster and 1/1335 cost to create a verified effective drug compound; and is high effective, low side effects and no toxicity

Based on 2024 Dollar value



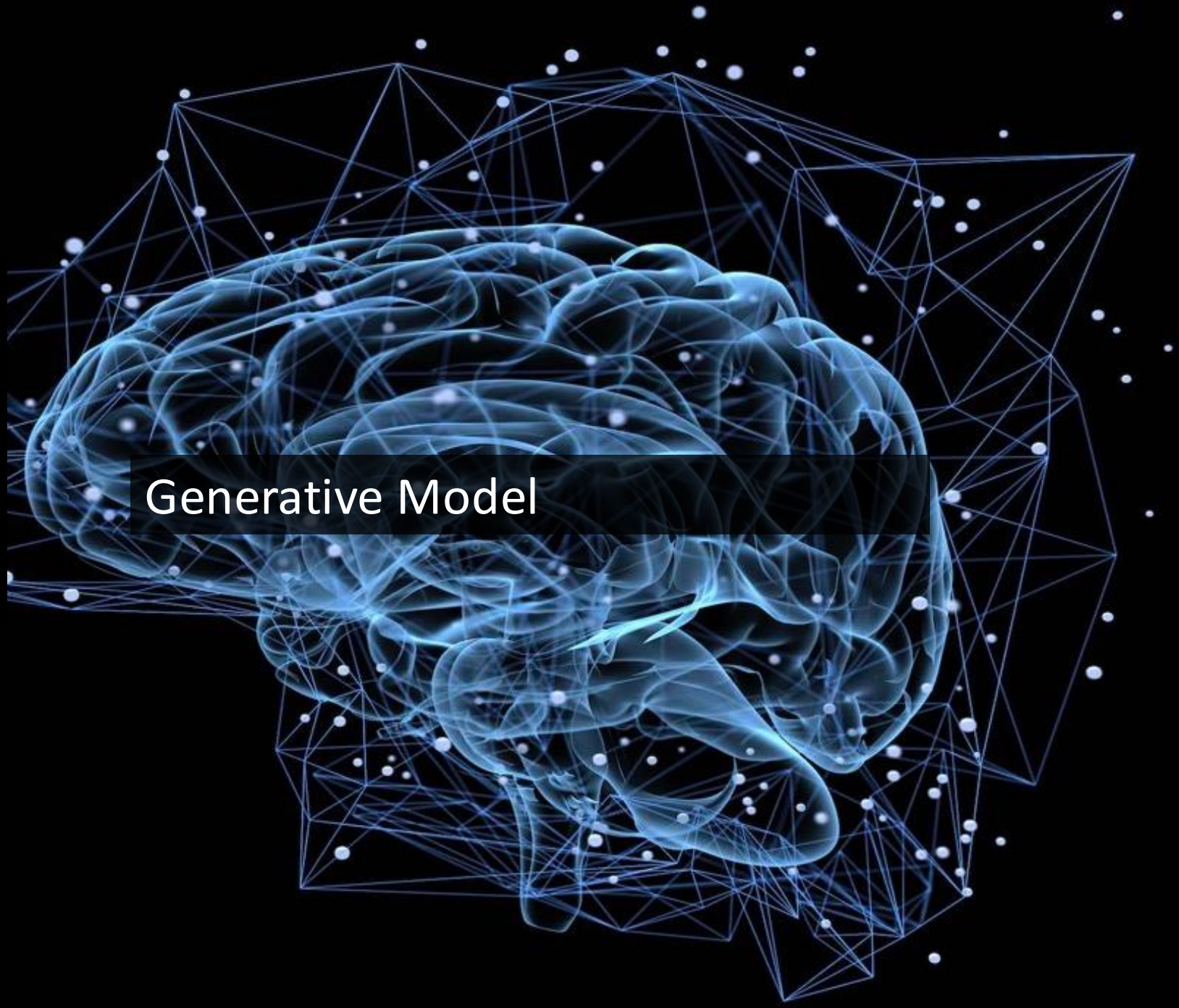

| | | Target-to-hit | Hit-to-lead | Lead optimization | Preclinical | Phase I | Phase II | Phase III | Submission to launch | |
|---|---|---|---|---|---|---|---|---|---|---|
| **General Pharmaceutical Companies** | **p(TS)** | 80% | 75% | 85% | 69% | 54% | 34% | 70% | 91% | |
| | **WIP in Launch** | 6.65 | 5.32 | 3.99 | 3.39 | 2.34 | 1.26 | 0.43 | 0.3 | 0.27 |
| | **Cycle Time (Year)** | 1.0 | 1.5 | 2.0 | 1.0 | 1.5 | 2.5 | 2.5 | 1.5 | |
| | **Cost per launch (capitalized, M)** | $37 | $66 | $164 | $60 | $108 | $127 | $122 | $19 | |
| | | $267 | | | $60 | $376 | | | | Total $703 |
| **Graphen Drugomics** | **p(TS)** | 90.9% | | | 80% | 75% | 60% | 80% | 95% | |
| | **WIP in Launch** | 1.1 | | | 1 | 0.8 | 0.6 | 0.36 | 0.3 | 0.27 |
| | **Cycle Time (Year)** | 0.167 | | | 1.0 | 1.5 | 2.5 | 2.5 | 1.5 | |
| | **Cost per launch (capitalized, M)** | $0.2 | | | $3 | $37 | $60 | $102 | $18 | |
| | | $0.2 | | | $3 | $217 | | | | Total $220 |

Graphen Drugomics's **Total cost saving** of **$483M** per 0.27 launch *($703M vs $220M)*

Graphen Drugomics's **IND-readies cost saving** of **$324M** per 0.27 launch *($327M vs $3.2M)*

➔ Potential Leads: 1/1335 of Cost & 27x faster. Up to IND-Ready: 1/102 of Cost & Time: 4.7x faster

COPYRIGHT © 2025 GRAPHEN DRUGOMICS, INC.


Generative Model

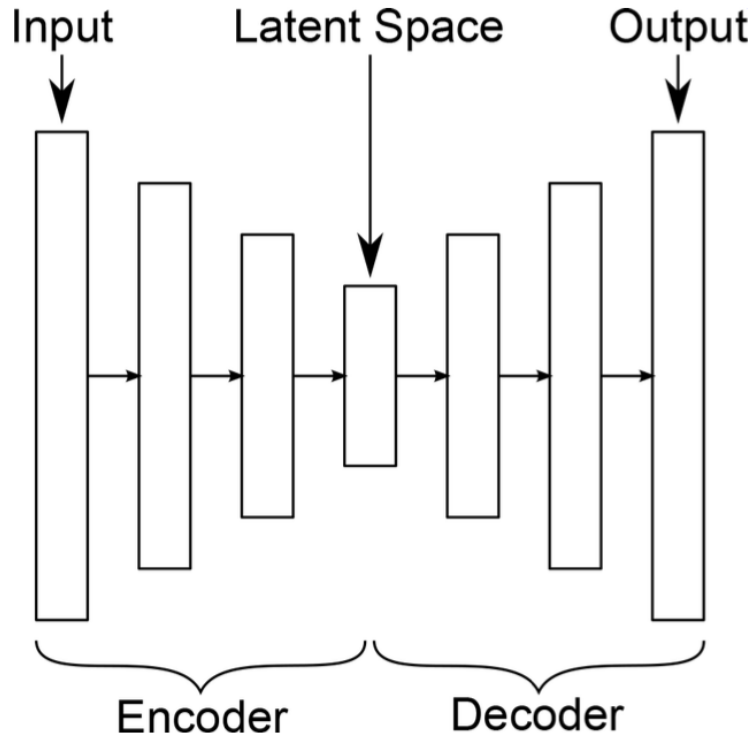
# Generative Models

- Generative models produce a sample as output.
- You might train the model on a library of photographs of cats, and it would learn to produce new images that look like cats.
- You might train it on a library of known drug molecules, and it would learn to generate new “drug-like” molecules for use as candidates in a virtual screen.
- Formally speaking, a generative model is trained on a collection of samples that are drawn from some (possibly unknown, probably very complex) probability distribution.
- Its job is to produce new samples from that same probability distribution.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Variational Autoencoders

- An autoencoder is a model that tries to make its output equal to its input.
- You train it on a library of samples and adjust the model parameters so that on every sample the output is as close as possible to the input.
- That sounds trivial. Can't it just learn to pass the input directly through to the output unchanged?
- If that were actually possible it would indeed be trivial, but autoencoders usually have architectures that make it impossible. Most often this is done by forcing the data to go through a bottleneck, as shown at right.
- For example, the input and output might each include 1,000 numbers, but in between would be a hidden layer containing only 10 numbers.
- This forces the model to learn how to compress the input samples. It must represent 1,000 numbers worth of information using only 10 numbers.

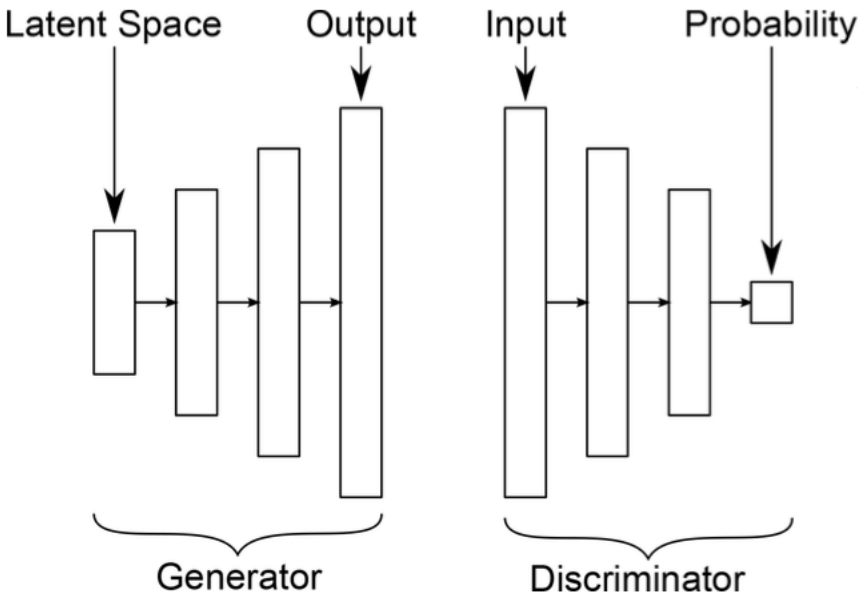




COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Generative Adversarial Networks

- A generative adversarial network (GAN) has much in common with a VAE.
- It uses the same exact decoder network to convert latent vectors into samples (except in a GAN, it is called the generator instead of the decoder).
- But it trains that network in a different way. It works by passing random vectors into the generator and directly evaluating the outputs on how well they follow the expected distribution.
- Effectively, you create a loss function to measure how well the generated samples match the training samples, then use that loss function to optimize the model.
- That sounds simple for a few seconds, until you think about it and realize it isn't simple at all. Could you write a loss function to measure how well an image resembles a cat?
- No, of course not! You wouldn't know where to begin. So, instead of asking you to come up with that loss function yourself, a GAN learns the loss function from the data.

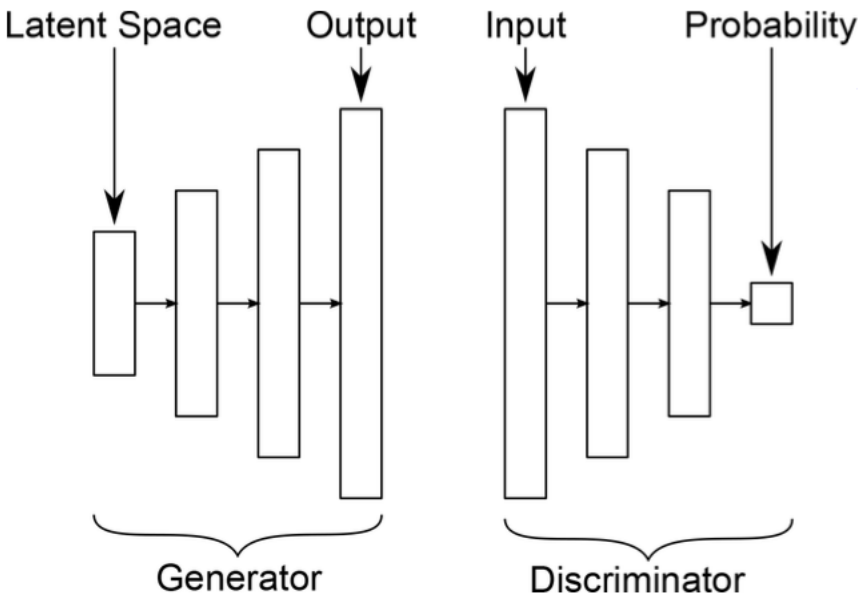




COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Generative Adversarial Networks

- As shown at the right, a GAN consists of two parts.
- The generator takes random vectors and generates synthetic samples.
- The second part, called the discriminator, tries to distinguish the generated samples from real training samples.
- It takes a sample as input and outputs a probability that this is a real training sample. It acts as a loss function for the generator.





COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

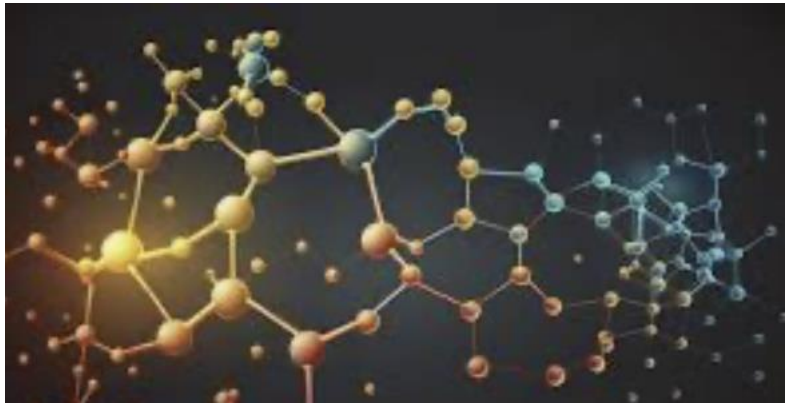
# Applications of Generative Models in the Life Science

- Broadly speaking, generative models bring a few superpowers to the table.
- First, they allow for a semblance of “creativity.”
- New samples can be generated according to the learned distribution.
- This allows for a powerful complement to a creative process that can tie into existing efforts in drug or protein design.
- Second, being able to model complex systems accurately with generative models could allow scientists to build an understanding of complex biological processes.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Generating New Ideas for Lead Compounds

- A major part of a modern drug discovery effort is coming up with new compounds.
- This is mostly done semiannually, with expert human chemists suggesting modifications to core structures.
- Often, this will involve projecting a picture of the current molecular series on a screen and having a room full of senior chemists suggest modifications to the core structure of the molecule.
- Some subset of these suggested molecules are actually synthesized and tested, and the process repeats until a suitable molecule is found or the program is dropped.
- This process has powerful advantages since it can draw upon the deep intuition of expert chemists who may be able to identify flaws with a potential structure (perhaps it resembles a compound they've seen before which caused unexplained liver failure in rats) that may not be easy to identify algorithmically.




COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Generating New Ideas for Lead Compounds

- At the same time, though, this process is very human-limited.
- There aren't that many talented and experienced senior chemists in the world, so the process can't scale outward.
- In addition, it makes it very challenging for a pharmaceutical division in a country that has historically lacked drug discovery expertise to bootstrap itself.
- A generative model of molecular structures could serve to overcome these limitations.
- If the model were trained on a suitable molecular representation, it might be able to rapidly suggest new alternative compounds.
- Access to such a model could help improve current processes by suggesting new chemical directions that may have been missed by human designers.
- It's worth noting that such design algorithms have serious caveats.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules

- We will train a VAE to generate new molecules.
- More specifically, it will output SMILES strings.
- This choice of representation has distinct advantages and disadvantages compared to some of the other representations we have discussed.
- On the one hand, SMILES strings are very simple to work with.
- Each one is just a sequence of characters drawn from a fixed alphabet.
- That allows us to use a very simple model to process them.
- On the other hand, SMILES strings are required to obey a complex grammar.
- If the model does not learn all the subtleties of the grammar, then most of the strings it produces will be invalid and not correspond to any molecule.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules - I

- The first thing we need is a collection of SMILES strings on which to train the model.
- Fortunately, MoleculeNet provides us with lots to choose from.
- For this example, we will use the MUV dataset.
- The training set includes 74,469 molecules of varying sizes and structures.
- Let's begin by loading it:

```
import deepchem as dc
tasks, datasets, transformers = dc.molnet.load_muv()
train_dataset, valid_dataset, test_dataset = datasets
train_smiles = train_dataset.ids
```


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules - II

- Next, we need to define the vocabulary our model will work with.
- What is the list of characters (or “tokens”) that can appear in a string?
- How long are strings allowed to be?
- We can determine these from the training data by creating a sorted list of every character that appears in any training molecule:

```
tokens = set()
for s in train_smiles:
  tokens = tokens.union(set(s))
tokens = sorted(list(tokens))
max_length = max(len(s) for s in train_smiles)
```


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules - III

- Now we need to create a model. What sort of architecture should we use for the
- encoder and decoder? This is an ongoing field of research.

- For this example, we will use DeepChem's AspuruGuzikAutoEncoder class, which implements a particular published model.

- It uses a convolutional network for the encoder and a recurrent network for the decoder. You can consult the original paper if you are interested in the details, but they are not necessary to follow the example.

- Also notice that we use ExponentialDecay for the learning rate. The rate is initially set to 0.001, then decreased by a little bit (multiplied by 0.95) after every epoch. This helps optimization to proceed more smoothly in many problems:

```
from deepchem.models.optimizers import ExponentialDecay
from deepchem.models.seqtoseq import AspuruGuzikAutoEncoder
batch_size = 100

batches_per_epoch = len(train_smiles)/batch_size
learning_rate = ExponentialDecay(0.001, 0.95, batches_per_epoch)
model = AspuruGuzikAutoEncoder(tokens, max_length, model_dir='vae',
                               batch_size=batch_size,
                               learning_rate=learning_rate)
```


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules - IV

- We are now ready to train the model.
- Instead of using the standard fit() method that takes a Dataset, AspuruGuzikAutoEncoder provides its own fit_sequences() method.
- It takes a Python generator object that produces sequences of tokens (SMILES strings in our case). Let's train for 50 epochs:

```
def generate_sequences(epochs):
  for i in range(epochs):
    for s in train_smiles:
      yield (s, s)

model.fit_sequences(generate_sequences(50))
```

- If everything has gone well, the model should now be able to generate entirely new molecules.
- We just need to pick random latent vectors and pass them through the decoder.
- Let's create a batch of one thousand vectors, each of length 196 (the size of the model's latent space).


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules - V

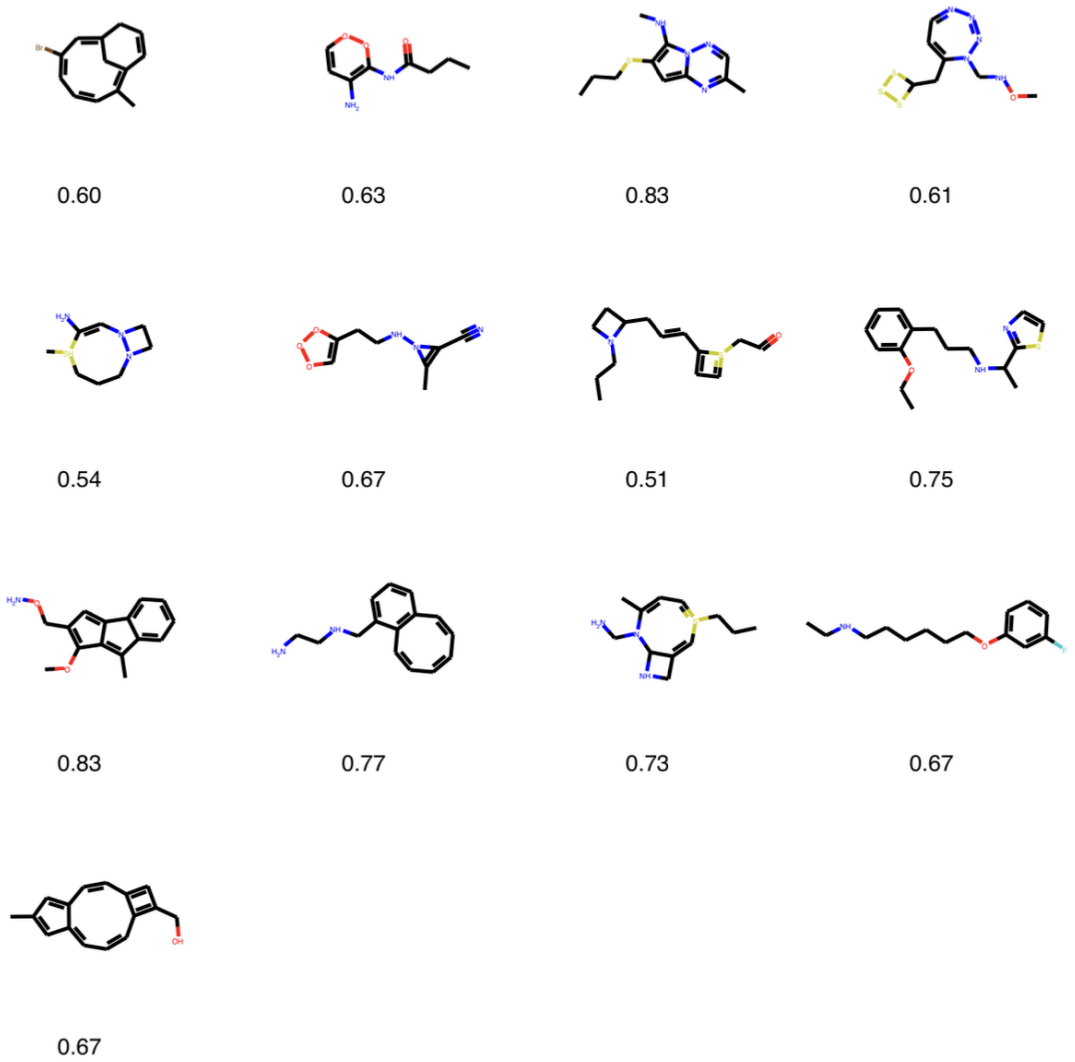
- As noted previously, not all outputs will actually be valid SMILES strings.
- In fact, only a small fraction of them are.
- Fortunately, we can easily use RDKit to check them and filter out the invalid ones:

```
import numpy as np
from rdkit import Chem
predictions = model.predict_from_embeddings(np.random.normal(size=(1000,196)))
molecules = []
for p in predictions:
  smiles = ''.join(p)
  if Chem.MolFromSmiles(smiles) is not None:
    molecules.append(smiles)
for m in molecules:
  print(m)
```


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training Generative Models to create new molecules - VI

- Several recent generative model publications use calculated molecular properties to determine which of the generated molecules to retain or discard.
- One of the more common methods for determining whether molecules are similar to known drugs, or “drug-like,” is known as the quantitative estimate of drug-likeness (QED).
- The QED metric, which was originally published by Bickerton and coworkers,1 scores molecules by comparing a set of properties calculated for each
- molecule with distributions of the same properties in marketed drugs.
- This score ranges between 0 and 1, with values closer to 1 being considered more drug-like.




 COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV. 

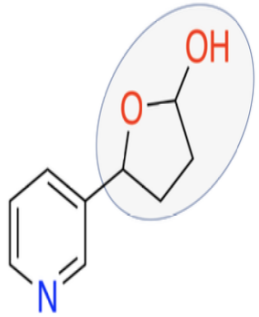
# Training Generative Models to create new molecules - VII

- While generative models provide an interesting means of producing ideas for new molecules, some key issues still need to be resolved to ensure their general applicability.
- The first is ensuring that the generated molecules will be chemically stable and that they can be physically synthesized.
- One current method to assess the quality of molecules produced by a generative model is to observe the fraction of the generated molecules that obey standard rules of chemical valence—in other words, ensuring that each carbon atom has four bonds, each oxygen atom has two bonds, each fluorine atom has one bond, and so on.
- These factors become especially important when decoding from a latent space with a SMILES representation.
- While a generative model may have learned the grammar of SMILES, there may be nuances that are still missing.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

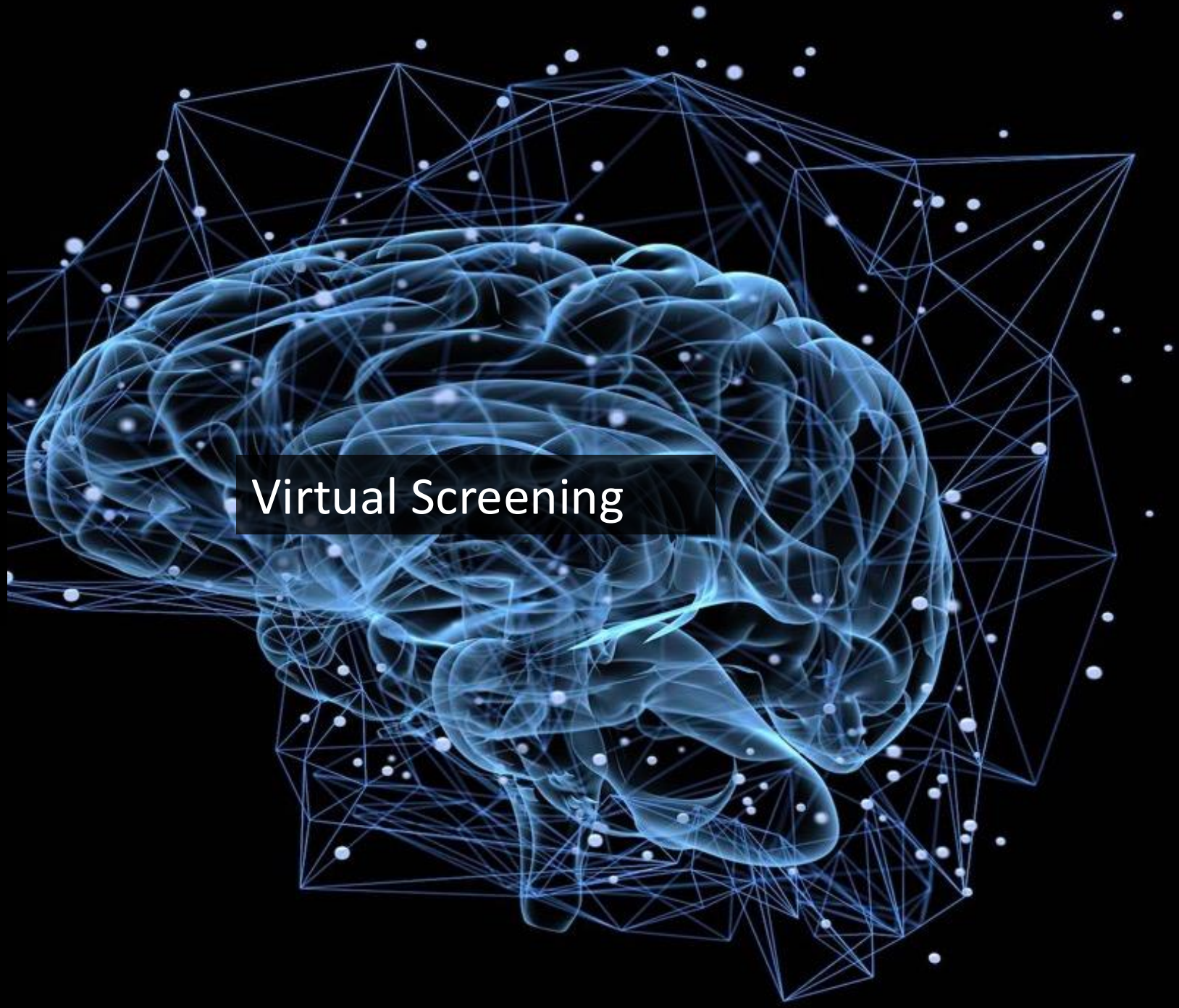
# Training Generative Models to create new molecules - VIII

- The fact that a molecule obeys standard rules of valence does not necessarily ensure that it will be chemically stable.
- In some cases, a generative model may produce molecules containing functional groups that are known to readily decompose.
- As an example, consider the molecule at the right. The functional group highlighted in the circle, known as a hemiacetal, is known to readily decompose.
- In practice, the probability of this molecule existing and being chemically stable is very small.
- There are dozens of chemical functionalities like this which are known to be unstable or reactive.
- When synthesizing molecules in a drug discovery project, medicinal chemists know to avoid introducing these functional groups.
- One way of imparting this sort of "knowledge" to a generative model is to provide a set of filters that can be used to postprocess the model output and remove molecules that may be problematic.



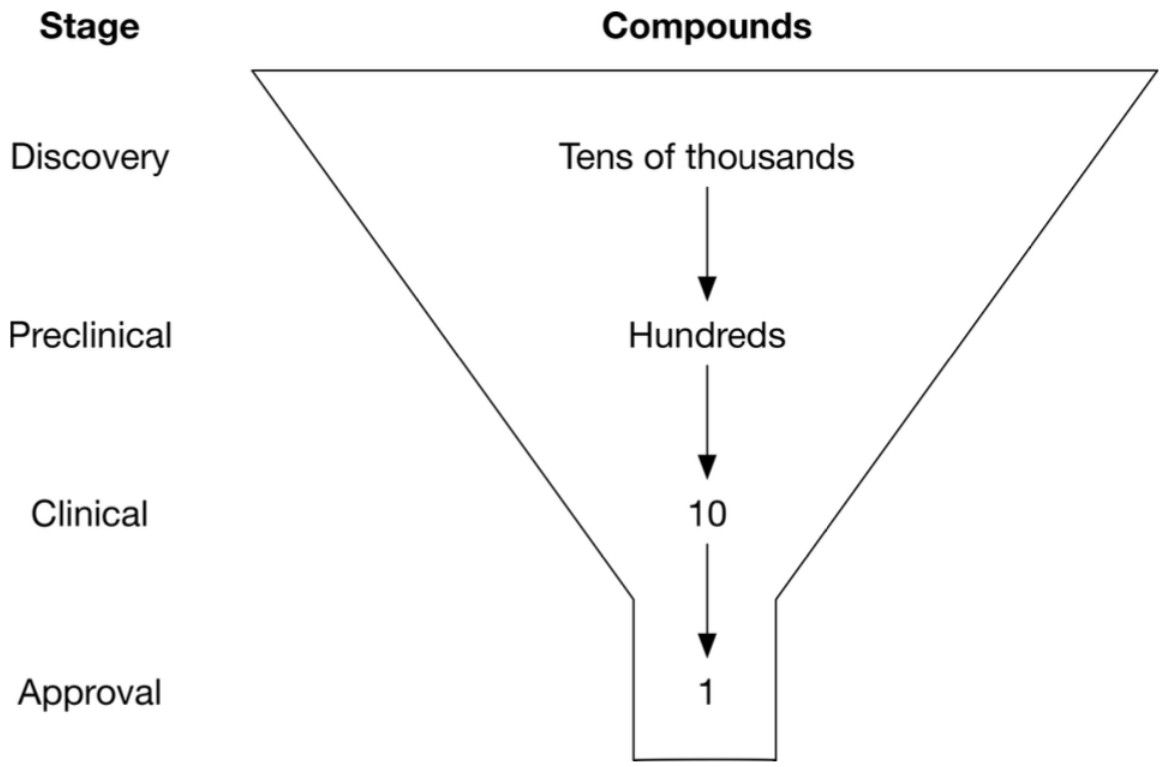

*A molecule containing an unstable group.*

 COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV. 


Virtual Screening

# Virtual Screening -- I

- Virtual screening can provide an efficient and cost-effective means of identifying starting points for drug discovery programs.
- Rather than carrying out an expensive, experimental high-throughput screen (HTS), we can use computational methods to virtually evaluate millions, or even tens of millions, of molecules.
- Virtual screening methods are often grouped into two categories, ***structure-based virtual screening*** and ***ligand-based virtual screening***.

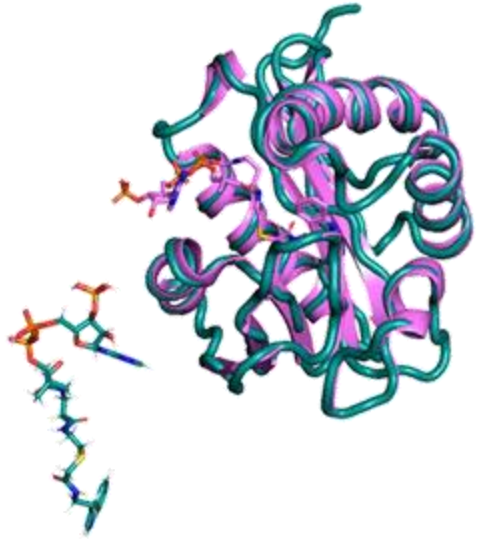




COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Structure-based Virtual Screening

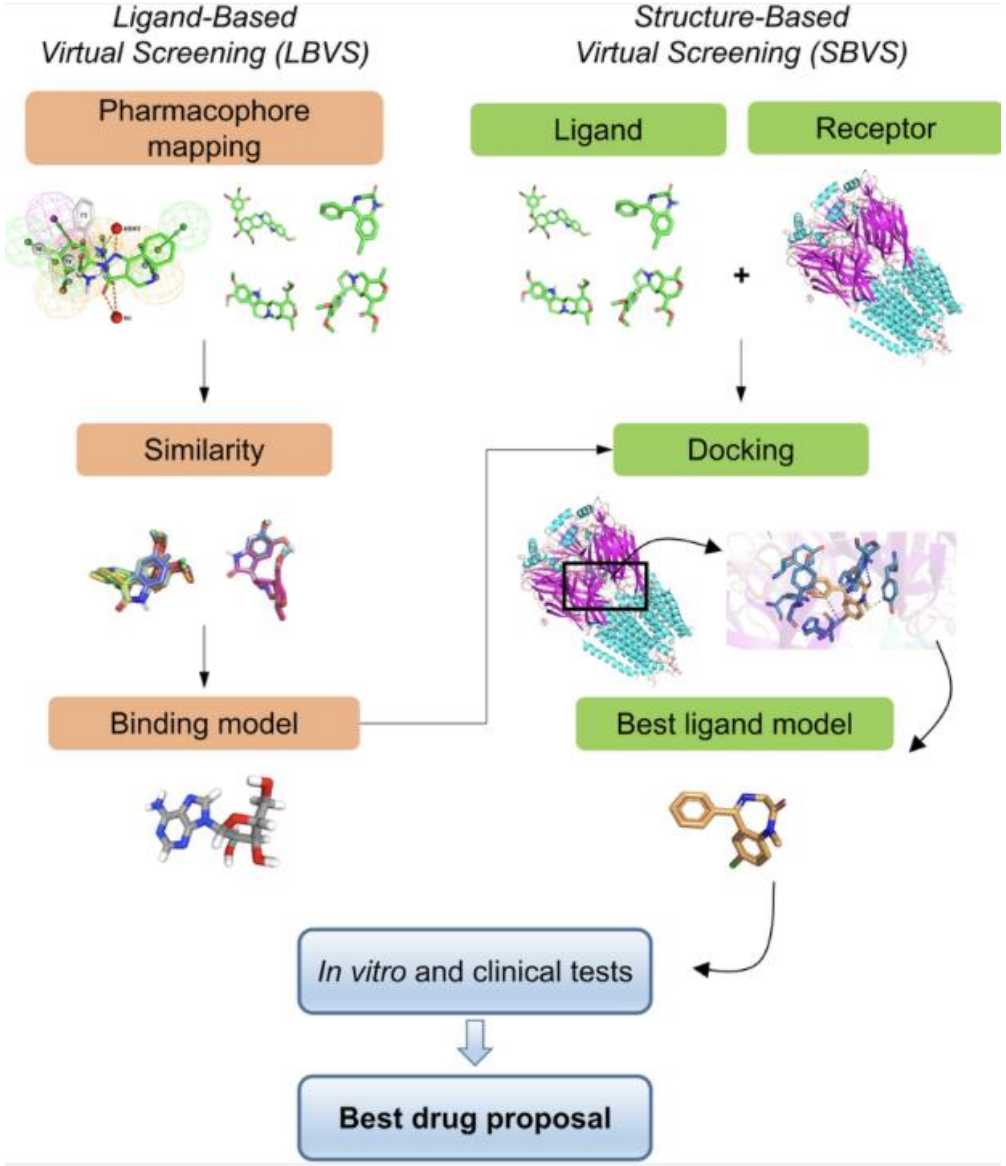
- In a structure-based virtual screen, computational methods are used to identify molecules that will optimally fit into a cavity, known as a binding site, in a protein.
- The binding of a molecule into the protein binding site can often inhibit the function of the protein.
- For instance, proteins known as enzymes catalyze a variety of physiological chemical reactions.
- By identifying and optimizing inhibitors of these enzymatic processes, scientists have been able to develop treatments for a wide range of diseases in oncology, inflammation, infection, and other therapeutic areas.




COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

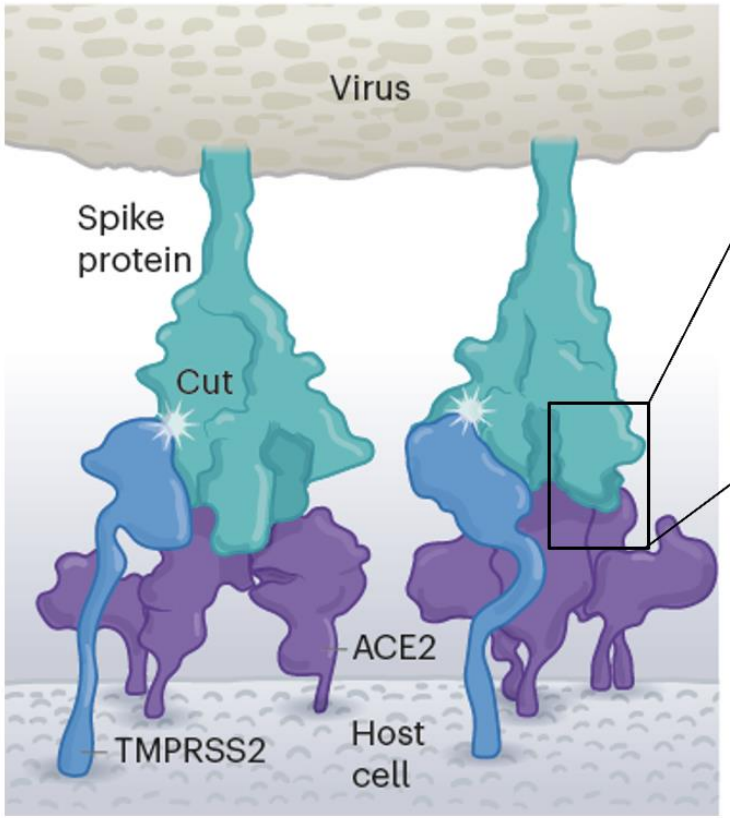
# Ligand-based Virtual Screening

- In a ligand-based virtual screen, we search for molecules that function similarly to one or more known molecules.
- We may be looking to improve the function of an existing molecule, to avoid pharmacological liabilities associated with a known molecule, or to develop novel intellectual property.
- A ligand-based virtual screen typically starts with a set of known molecules identified through any of a variety of experimental methods.
- Computational methods are then used to develop a model based on experimental data, and this model is used to virtually screen a large set of molecules to find new chemical starting points.

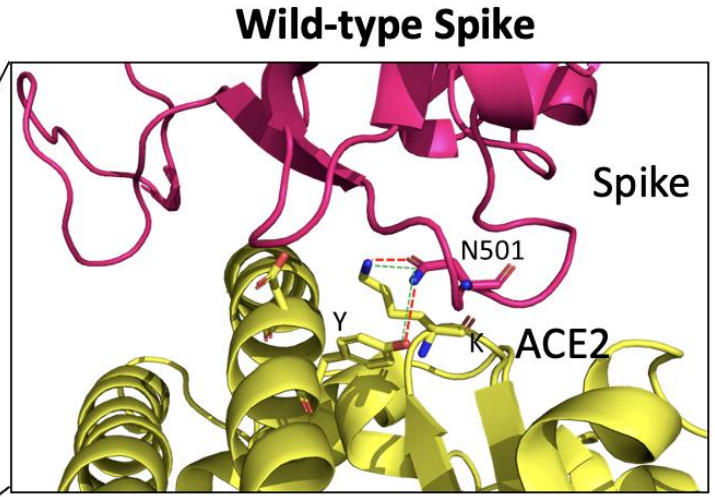


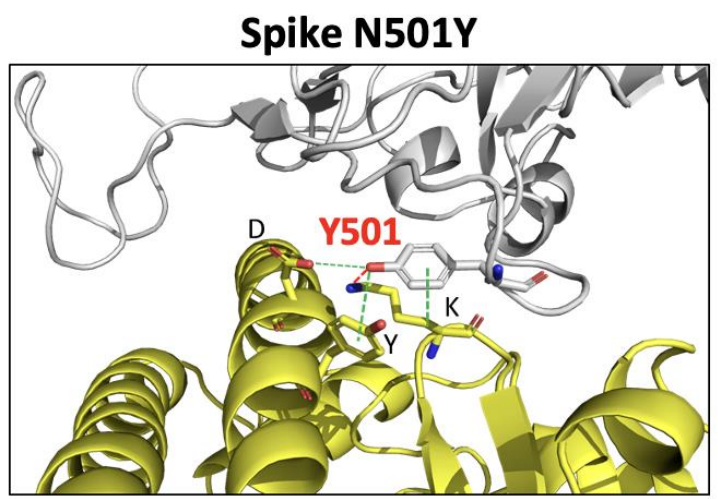

 COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV. 

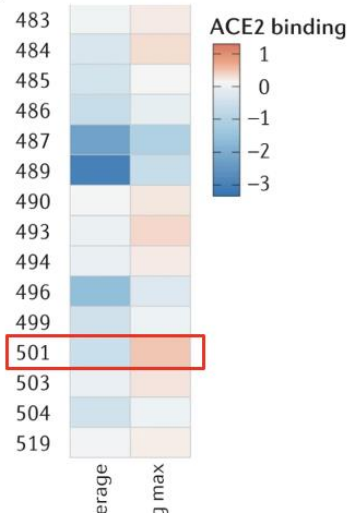
# Example: COVID-19 Alpha Variant




Source: Janet Iwasa, Univ. Utah;
Graphic: Nik Spencer/Nature



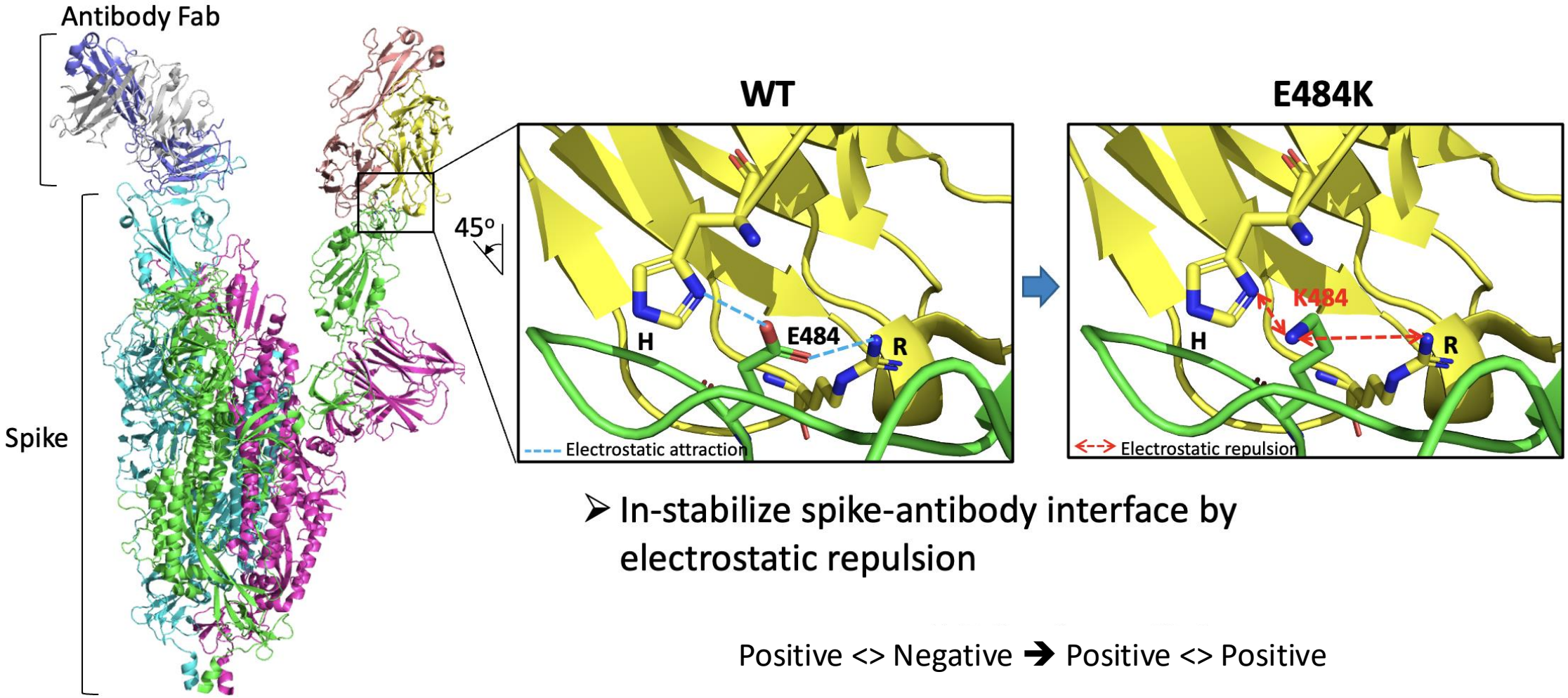




- Protein structure reveals that N501Y mutation enhances the binding ability to ACE2[1]
- N501Y has been shown experimentally to result in one of the highest increases in ACE2 affinity conferred by a single RBD mutation[2]


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Example: COVID-19 Beta Variant



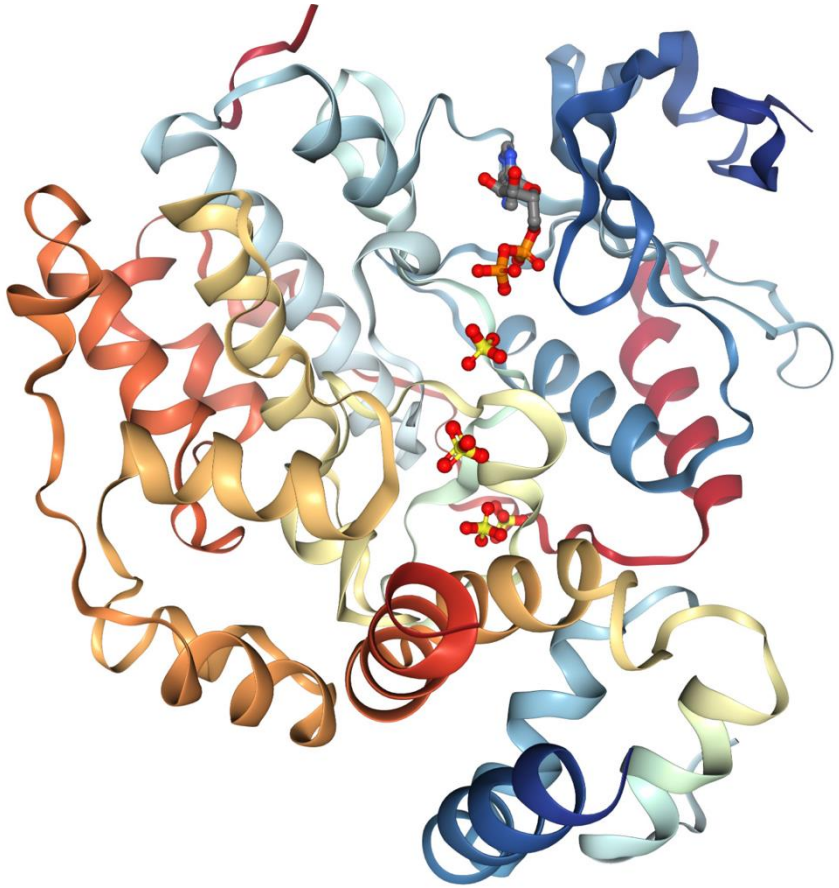

- In-stabilize spike-antibody interface by electrostatic repulsion

Positive <> Negative ➔ Positive <> Positive


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

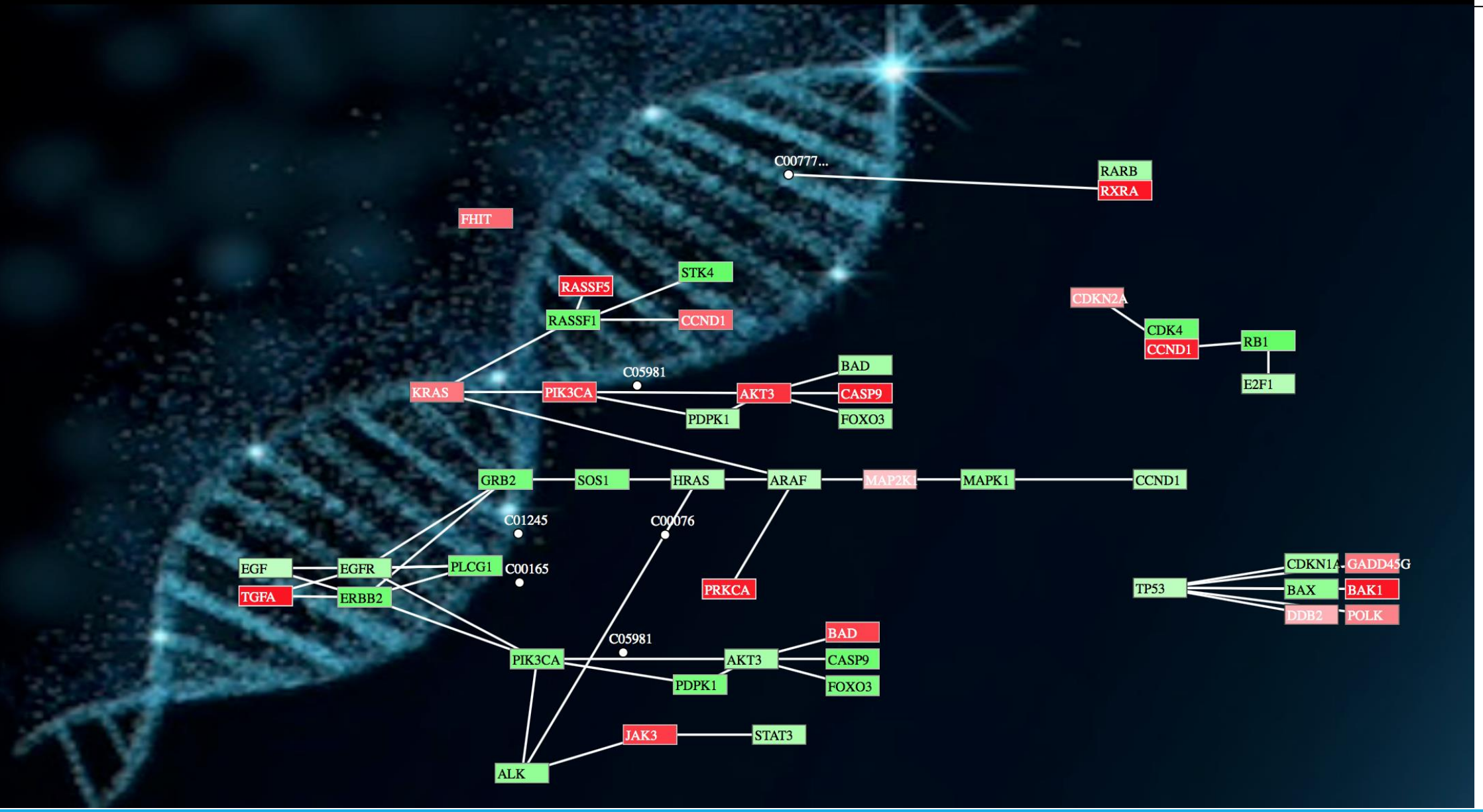
# Example of Predicting Molecules to inhibit a protein - I

- We will build a graph convolution model to predict the ability of molecules to inhibit a protein known as ERK2.
- This protein, also known as mitogen-activated protein kinase 1, or MAPK1, plays an important role in the signaling pathways that regulate how cells multiply.
- ERK2 has been implicated in a number of cancers, and ERK2 inhibitors are currently being tested in clinical trials for non-small-cell lung cancer and melanoma (skin cancer).




COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Graphen Proteogenomics Pathway Analyzer -- Non-small cell lung cancer

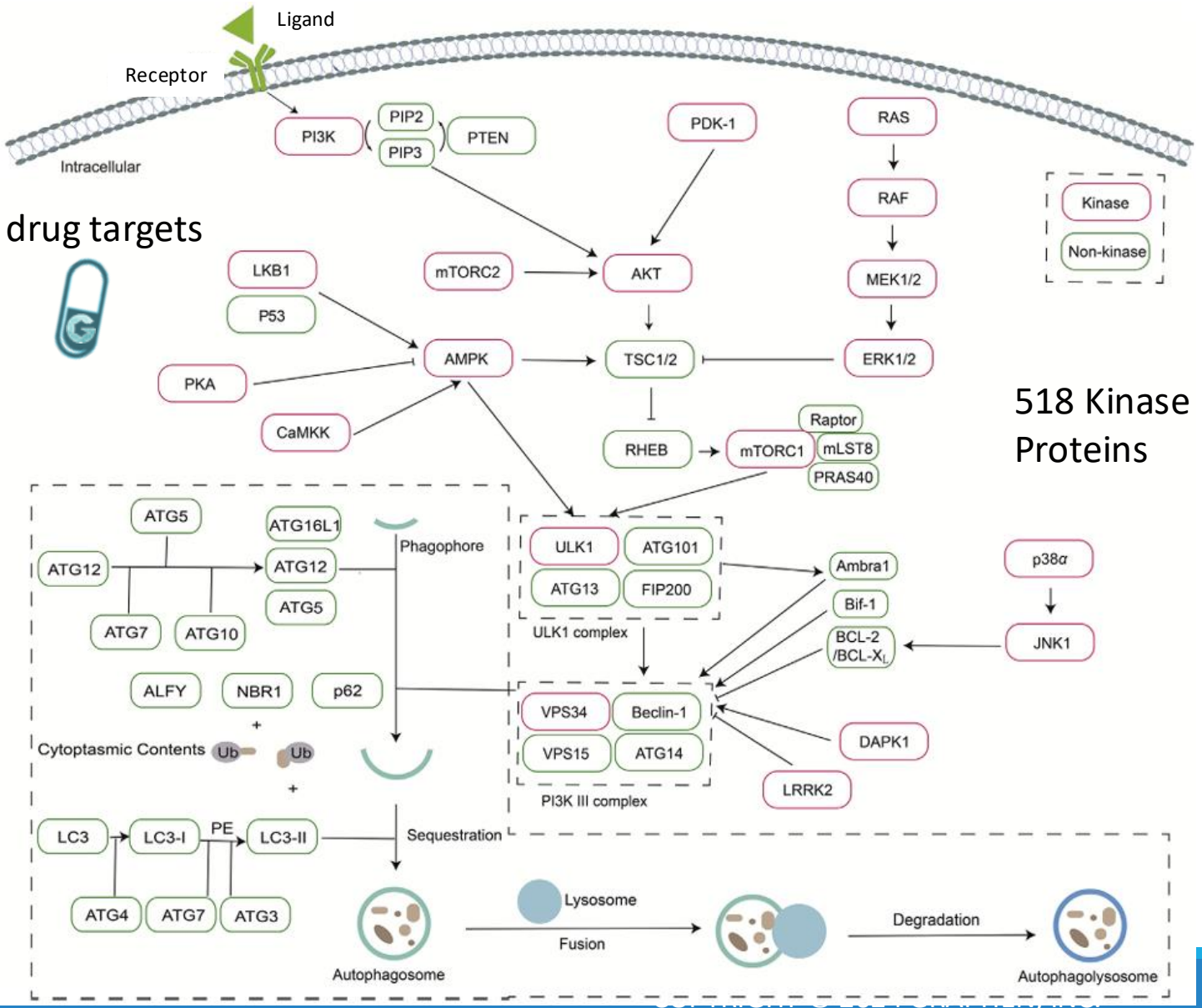



 COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV. 

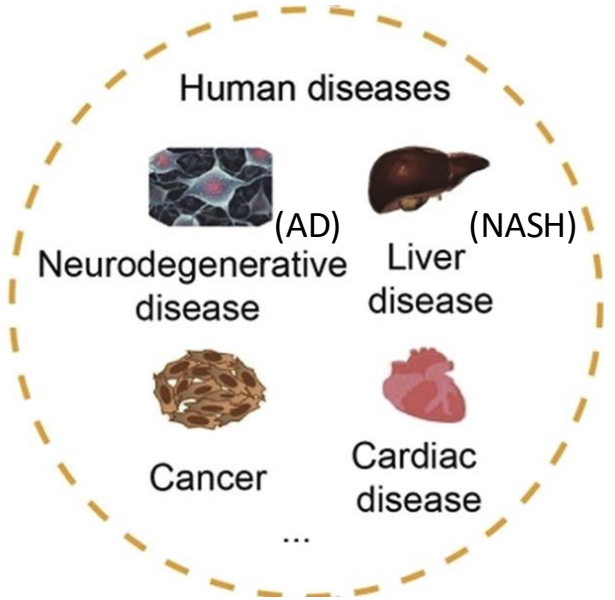
# Kinases are pivotal in multi-disease pathways

Graphen's drug targets

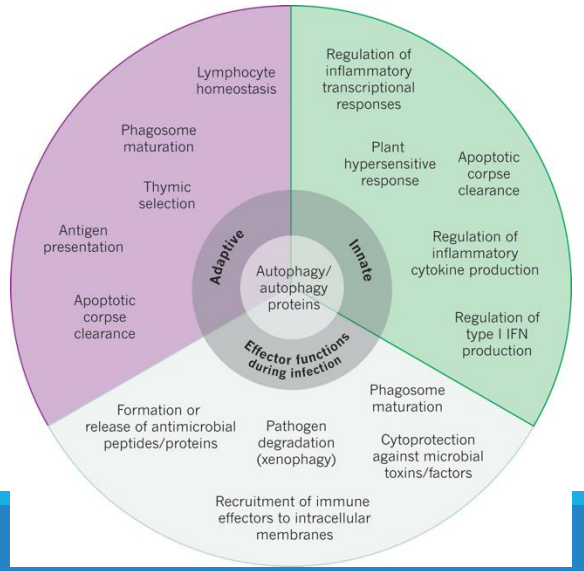
*Quickly design as many as Kinase drugs as possible.*




518 Kinase Proteins



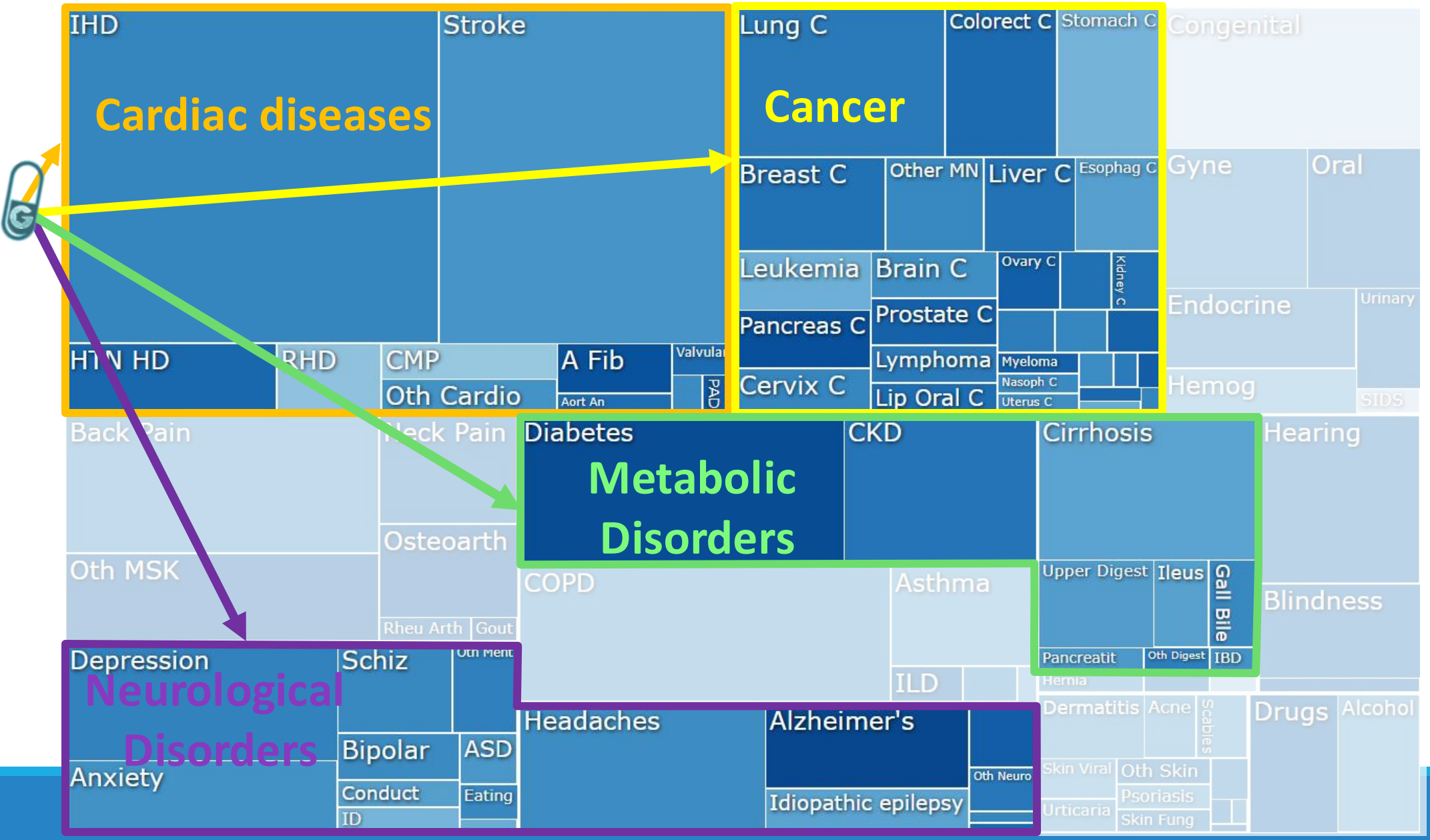

Immune disorder



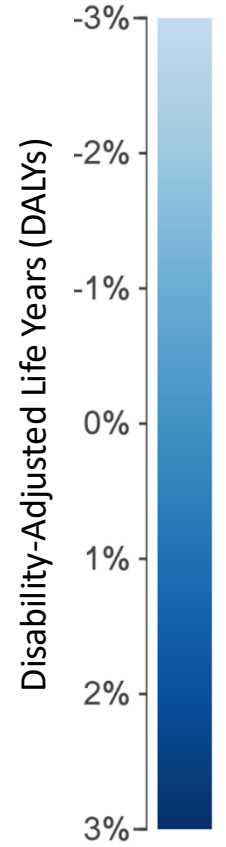

Xiang H., et.al. Targeting autophagy-related protein kinases for potential therapeutic purpose. Acta Pharm Sin B. 2020 Apr;10(4):569-581.

Levine, B., et.al. Autophagy in immunity and inflammation. Nature 469, 323–335 (2011).

# Graphen's drugs can potentially cover around 70% of disease fields




Annual % change (2010 to 2019) DALYs/100,000

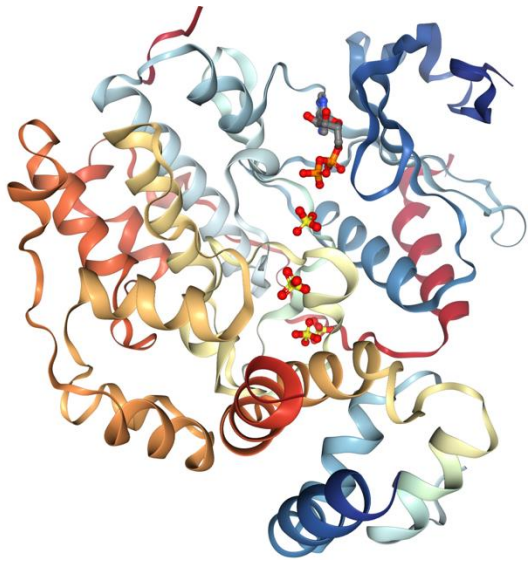



Data from IHME (1990 to 2019)

IHME
THE INSTITUTE FOR HEALTH METRICS AND EVALUATION

# Example of Predicting Molecules to inhibit a protein - II

- We will train the model to distinguish a set of ERK2 active compounds from a set of decoy compounds.
- The active and decoy compounds are derived from the DUD-E database, which is designed for testing predictive models.
- In practice, we would typically obtain active and inactive molecules from the scientific literature, or from a database of biologically active molecules such as the ChEMBL database from the European Bioinformatics Institute (EBI).
- In order to generate the best model, we would like to have decoys with property distributions similar to those of our active compounds.
- Let's say this was not the case and the inactive compounds had lower molecular weight than the active compounds.
- In this case, our classifier might be trained simply to separate low molecular weight compounds from high molecular weight compounds.

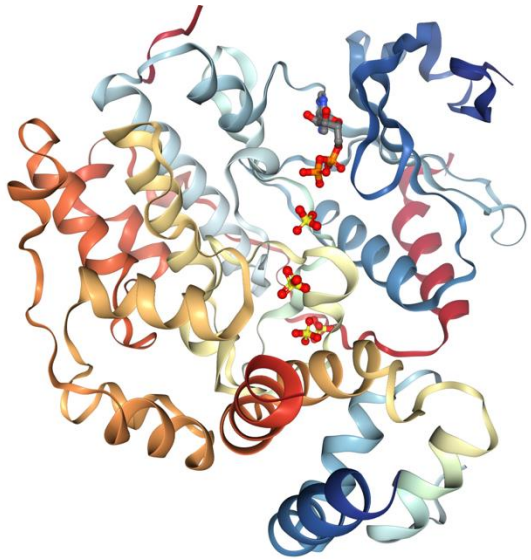



COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Example of Predicting Molecules to inhibit a protein - III

- In order to better understand the dataset, let's examine a few calculated properties of our active and decoy molecules.

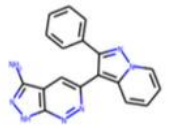
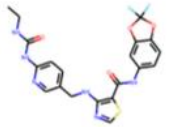
- To build a reliable model, we need to ensure that the properties of the active molecules are similar to those of the decoy molecules.



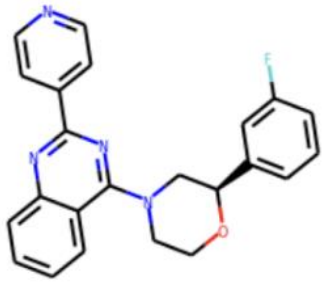

COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Example of Predicting Molecules to inhibit a protein - IV

Predicted False Negative Modules

| | neg | pos | active | SMILES | Mol |
|---|---|---|---|---|---|
| 4 | 0.723421 | 0.27658 | 1 | c1ccc(cc1)c2c(c3ccccn3n2)c4cc5c(n[nH]c5nn4)N |  |
| 5 | 0.910040 | 0.08996 | 1 | CCNC(=O)Nc1ccc(cn1)CNc2c(scn2)C(=O)Nc3ccc4c(c3)OC(O4)(F)F |  |

Predicted False Positive Module

| | neg | pos | active | SMILES | Mol |
|---|---|---|---|---|---|
| 296 | 0.564975 | 0.435025 | 0 | c1ccc2c(c1)c(nc(n2)c3ccncc3)N4CCO[C@@H](C4)c5cccc(c5)F |  |


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Training a Predictive Model

- Let's define a function to create a GraphConvModel.
- In this case, we will be creating a classification model.
- Since we will apply the model later on a different dataset, it's a good idea to create a directory in which to store the model.

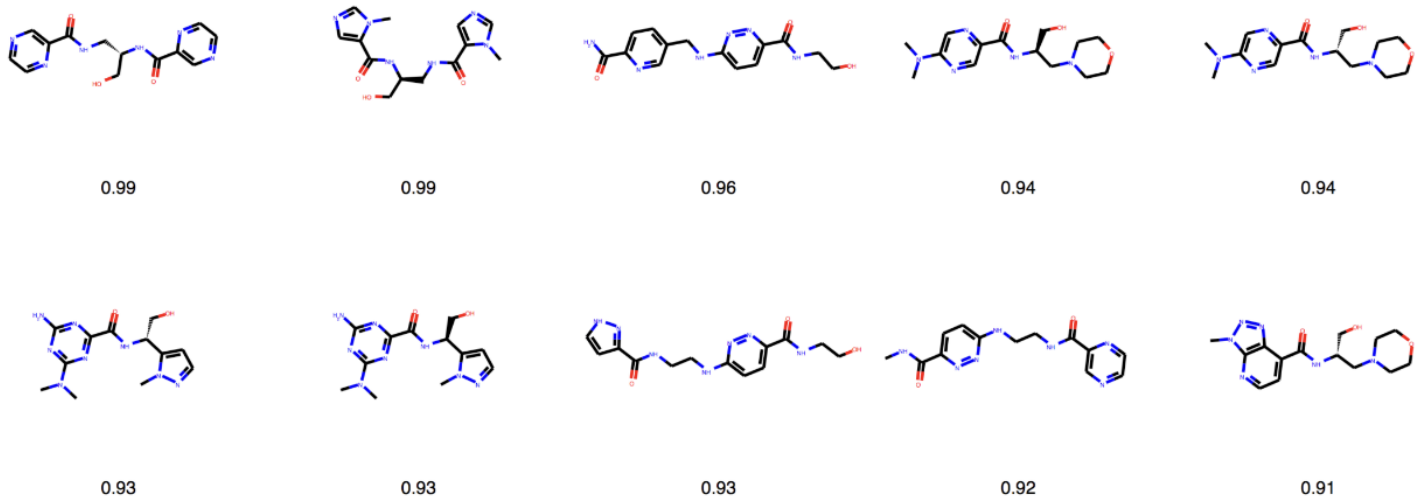
```
def generate_graph_conv_model():
batch_size = 128
model = GraphConvModel(1, batch_size=batch_size,
mode='classification',
model_dir="/tmp/mk01/model_dir")
return model
```

- We will create training and test sets to evaluate the model's performance.
- In this case, we will use the Random
- Splitter (DeepChem offers a number of other splitters too, such as the ScaffoldSplitter, which divides the dataset by chemical scaffold, and the ButinaSplitter, which first clusters the data then splits the dataset so that different clusters end up in the training and test sets)
-


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Filtering from available compounds

- The GraphConvMdel we created can now be used to search the set of commercially available compounds we just filtered.




- Indeed, many of the molecules are very similar and might end up being redundant in our screen. One way to be more efficient would be to cluster the molecules and only screen the highest-scoring molecule in each cluster.

- RDKit has an implementation of the Butina clustering method, one of the most highly used methods in cheminformatics.

- In the Butina clustering method, we group molecules based on their chemical similarity, which is calculated using a comparison of bit vectors (arrays of 1 and 0), also known as chemical fingerprints that represent the presence or absence of patterns of connected atoms in a molecule.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Clustering the molecules

- A small amount of code is necessary to cluster a set of molecules.
- The only parameter required for Butina clustering is the cluster cutoff.
- If the Tanimoto similarity of two molecules is greater than the cutoff, the molecules are put into the same cluster.
- If the similarity is less than the cutoff, the molecules are put into different clusters.

```
def butina_cluster(mol_list, cutoff=0.35):
    fp_list = [
        rdmd.GetMorganFingerprintAsBitVect(m, 3, nBits=2048)
        for m in mol_list]
    dists = []
    nfps = len(fp_list)

for i in range(1, nfps):
    sims = DataStructs.BulkTanimotoSimilarity(
        fp_list[i], fp_list[:i])
    dists.extend([1 - x for x in sims])
mol_clusters = Butina.ClusterData(dists, nfps, cutoff,
                                  isDistData=True)
cluster_id_list = [0] * nfps
for idx, cluster in enumerate(mol_clusters, 1):
    for member in cluster:
        cluster_id_list[member] = idx
return cluster_id_list
```


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Choose representative compounds in clusters

- Before clustering, we will create a new dataframe with only the 100 top-scoring molecules.

```
best_100_df = combo_df.head(100).copy()
```

- We can then create a new column containing the cluster identifier for each compound:

```
best_100_df["Cluster"] = butina_cluster(best_100_df.Mol)
best_100_df.head()
```

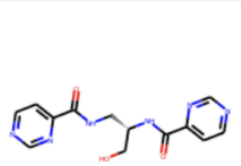
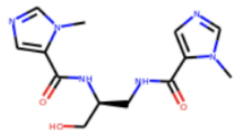
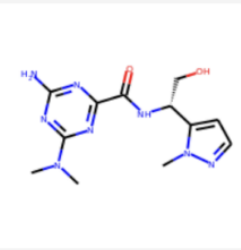
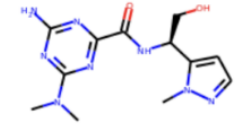
- We now see that in addition to the SMILES string, molecule name, and predicted values, we also have a cluster identifier as in the right figure.

- We can use the Pandas unique function to determine that we have 55 unique clusters:

```
len(best_100_df.Cluster.unique())
```

- Ultimately, we would like to purchase these compounds and screen them experimentally. In order to do this, we need to save a CSV file listing the molecules we plan to

```
best_cluster_rep_df = best_100_df.drop_duplicates("Cluster")
```

| | SMILES | Name | Val | Neg | Pos | Mol | Cluster |
|---|---|---|---|---|---|---|---|
| 63669 | O=C(NC[C@@H](CO)NC(=O)c1ccncn1)c1ccncn1 | ZINC000681745616 | 0 | 0.438595 | 0.561404 |  | 55 |
| 55121 | Cn1cncc1C(=O)NC[C@@H](CO)NC(=O)c1cncn1C | ZINC000644062250 | 0 | 0.481628 | 0.518372 |  | 54 |
| 38671 | CN(C)c1nc(N)nc(C(=O)N[C@H](CO)c2ccnn2C)n1 | ZINC000566403331 | 0 | 0.501487 | 0.498513 |  | 26 |
| 38672 | CN(C)c1nc(N)nc(C(=O)N[C@@H](CO)c2ccnn2C)n1 | ZINC000566403338 | 0 | 0.501487 | 0.498513 |  | 26 |


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Summary -- I

- We have followed the steps of a ligand-based virtual screening work-flow.
- We used deep learning to build a classification model that was capable of distinguishing active from inactive molecules.
- The process began with evaluating our training data and ensuring that the molecular weight, LogP, and charge distributions were balanced between the active and decoy sets.
- Once we'd made the necessary adjustments to the chemical structures of the decoy molecules, we were ready to build a model.
- The first step in building the model was generating a set of chemical features for the molecules being used.
- We used the DeepChem GraphConv featurizer to generate a set of appropriate chemical features.
- These features were then used to build a graph convolution model, which was subsequently used to predict the activity of a set of commercially available molecules.
- In order to avoid molecules that could be problematic in biological assays, we used a set of computational rules encoded as SMARTS patterns to identify molecules containing chemical functionality previously known to interfere with assays or create subsequent liabilities.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Summary -- II

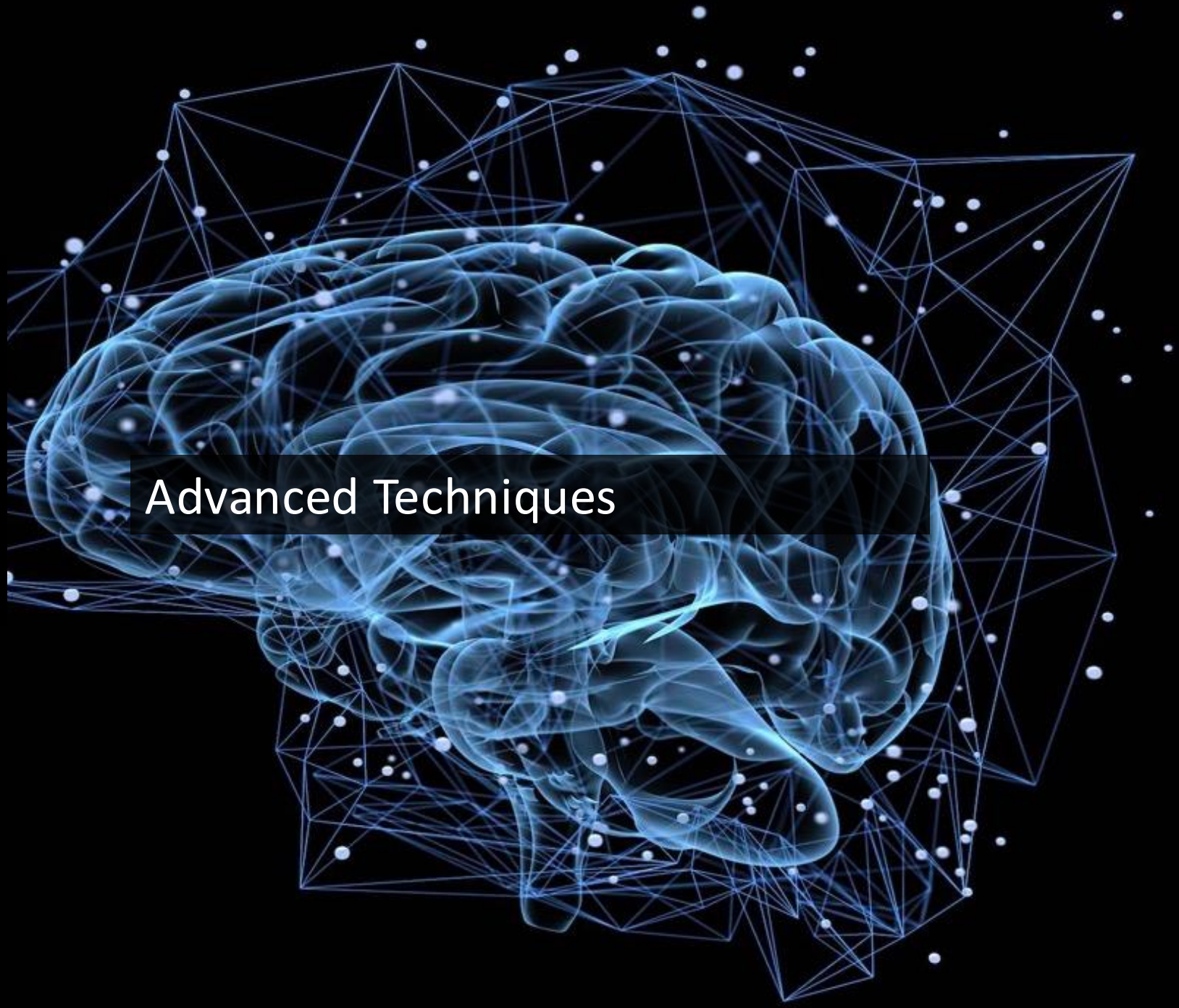
- With our list of desired molecules in hand, we are in a position to test these molecules in biological assays.
- Typically the next step in our workflow would be to obtain samples of the chemical compounds for testing.
- If the molecules came from a corporate compound collection, a robotic system would collect the samples and prepare them for testing.
- If the molecules were purchased from commercial sources, additional weighing and dilution with buffered water or another solvent would be necessary.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.

# Summary -- III

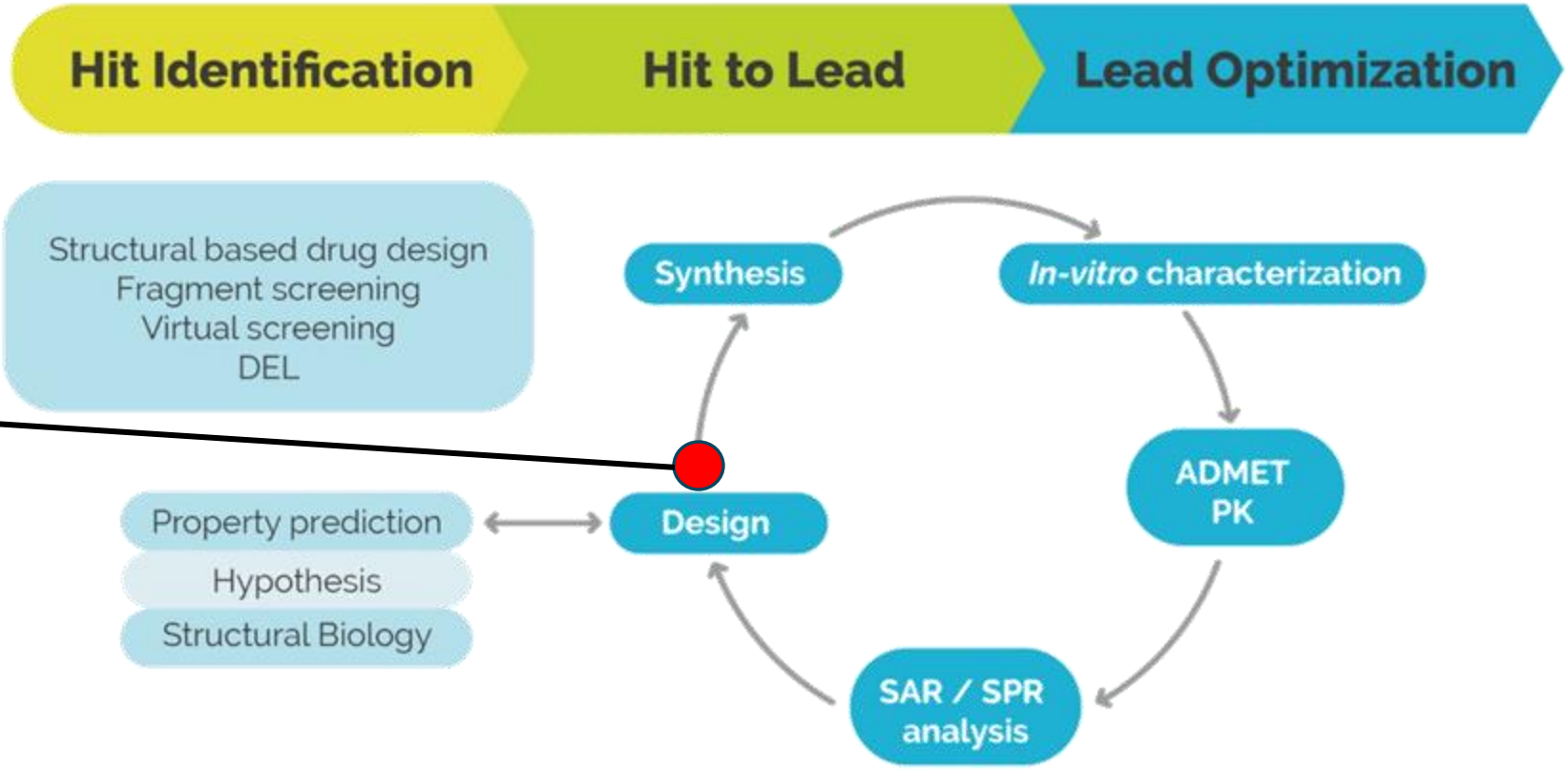
- Once the samples are prepared, they are tested in biological assays.
- These assays can cover a wide range of endpoints, ranging from inhibiting bacterial growth to preventing the proliferation of cancer cells.
- While the testing of these molecules is the final step in our virtual screening exercise, it is far from the end of the road for a drug discovery project.
- Once we have run the initial biological assay on the molecules we identified through virtual screening, we analyze the results of the screen.
- If we find experimentally active molecules, we will typically identify and test other similar molecules that will enable us to understand the relationships between different parts of the molecule and the biological activity that we are measuring.
- This optimization process often involves the synthesis and testing of hundreds or even thousands of molecules to identify those with the desired combination of safety and biological activity.


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.


Advanced Techniques

# AI in BioMedical

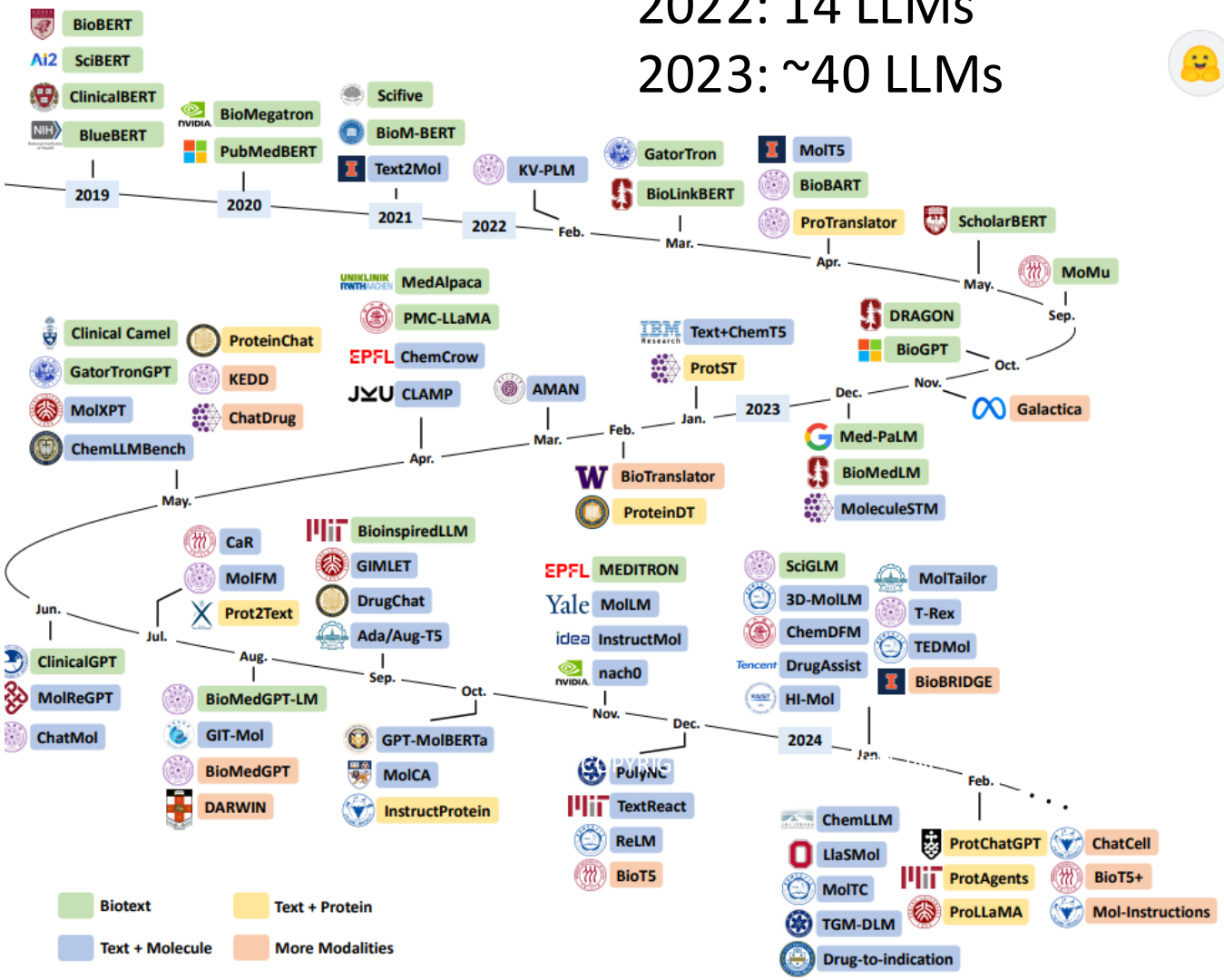
Traditional Drug development




**Before Design, needs Target Structure**

# LLMs in Bio-Medicine Field

2022: 14 LLMs
2023: ~40 LLMs




🤗 Hugging Face open source:

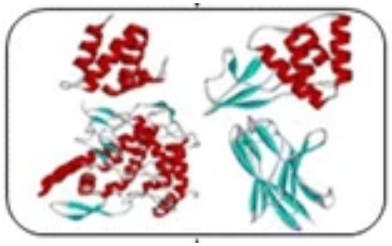
- **ProtGPT2**[1]
  - 738M params
- **AntibodyGPT**[2]
  - 151M, 764M, 2.7B, 6.4B
- **ChemGPT**[3]
  - 4.7M, 19M, 1.2B
- **DNAGPT**[4]
  - 100M, 3B

1. Ferruz N, Schmidt S, Höcker B. ProtGPT2 is a deep unsupervised language model for protein design. Nat Commun. 2022;13(1):4348. Published 2022 Jul 27. doi:10.1038/s41467-022-32007-7
2. https://huggingface.co/AntibodyGeneration
3. https://huggingface.co/ncfrey
4. https://github.com/TencentAILabHealthcare/DNAGPT

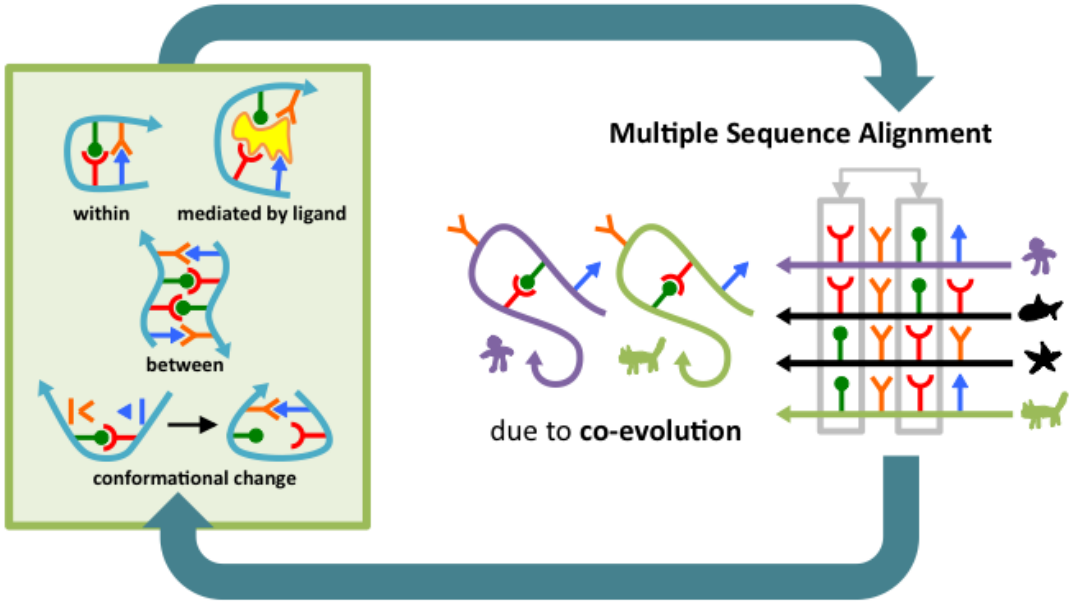
# The logics of Protein Structure Model



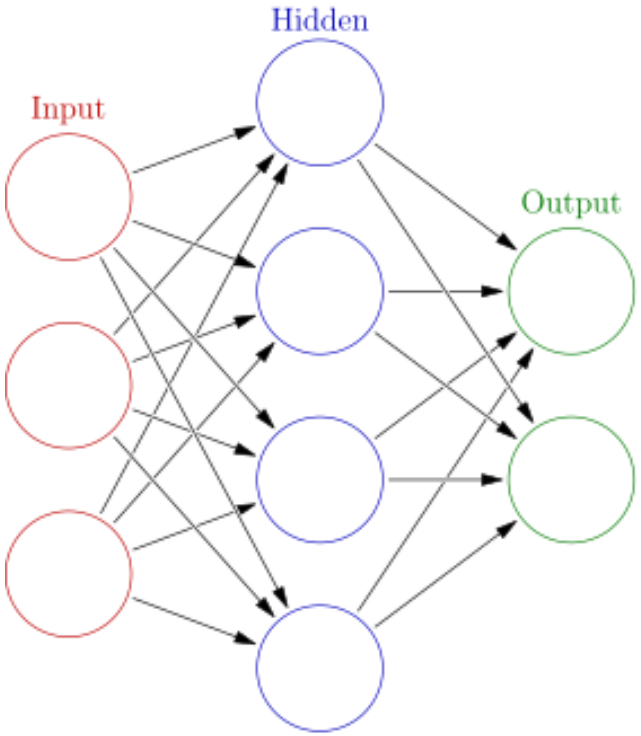
## Template for similar secondary structure features



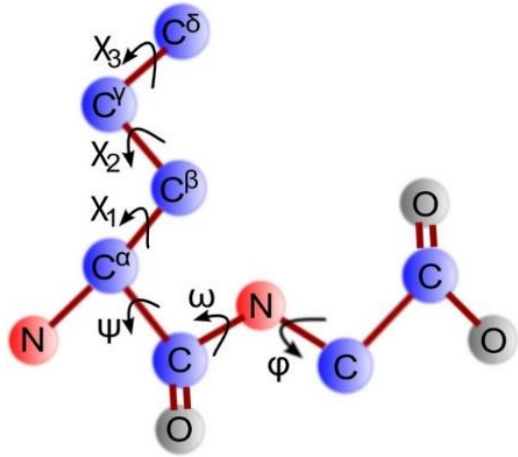
## MSA for co-evolution features in different species



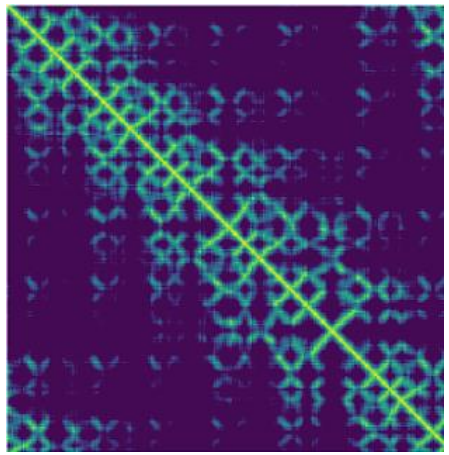

## Model training




## Orientation



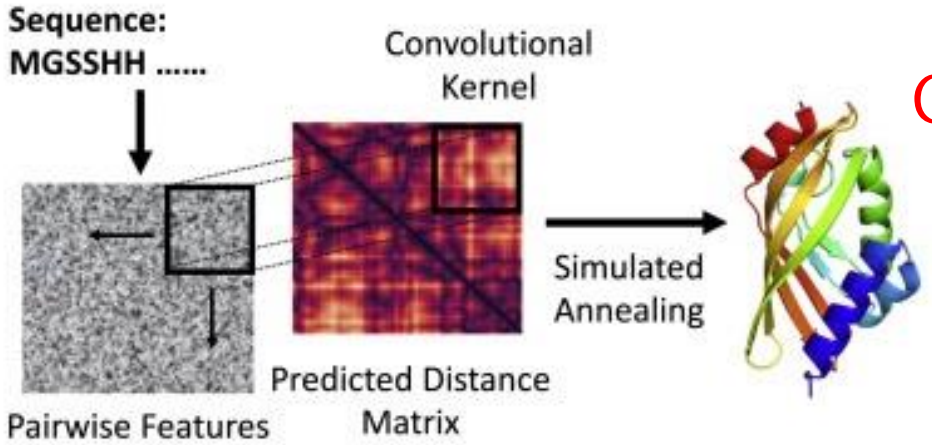

## Distance Matrix



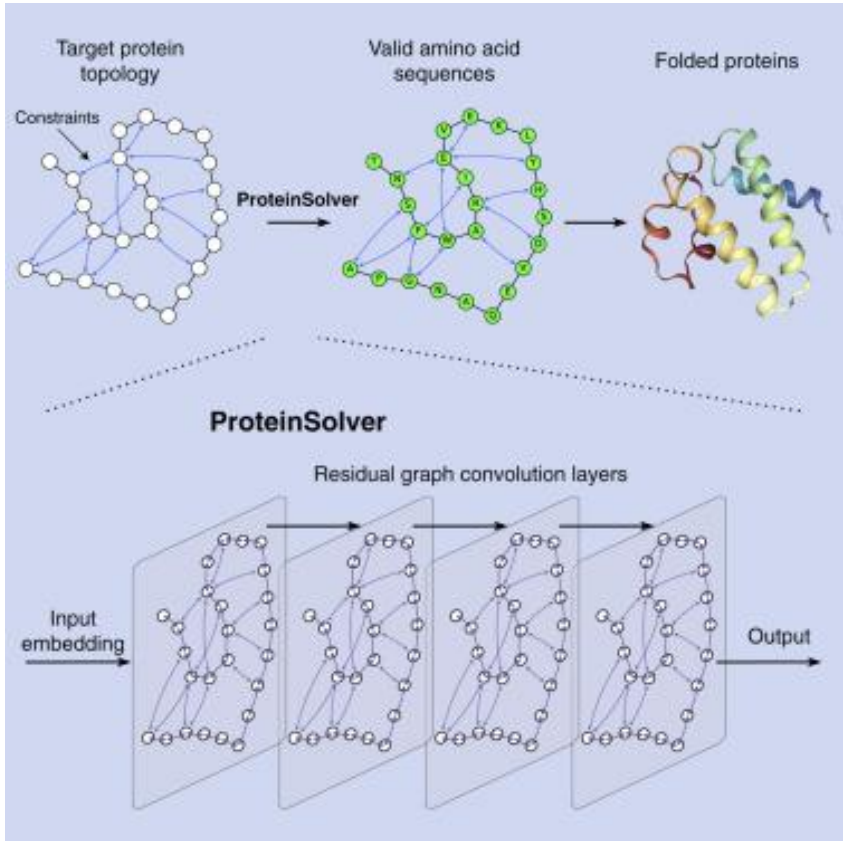
# Deep Learning for Protein Structure Model



## Convolution Neural Network

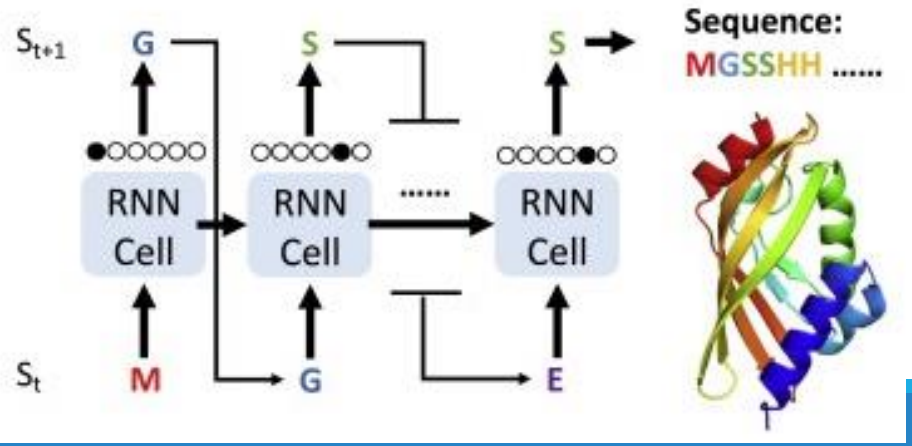



## Graph-based Convolution Network



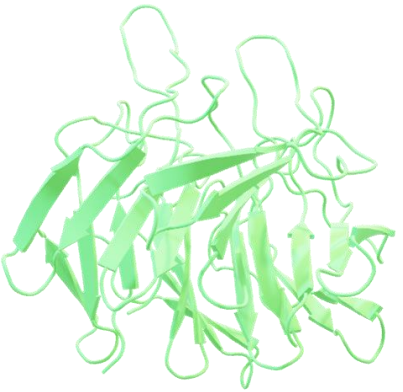

- Order problem
- Global problem

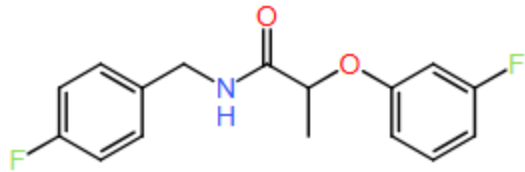
## Recurrent Neural Network



# Protein to Drug Development

- Protein is crucial for drug development, and the AlphaFold database has deciphered the structures of over 200 million proteins.
- AlphaFold can effectively interpret binding between different structures but cannot quantify the binding affinity.
- In drug development, aside from efficacy, approximately 20% of clinical trial failures are due to safety issues. AlphaFold cannot assess the safety and manufacturability of drugs.
- AlphaFold completes the first step in computational drug science, but there is still a long journey ahead...





Protein Structure Model

# Alphafold Evalution

COLUMBIA UNIVERSITY

For 50 years CASP competition:

Before : 50%

Alphafold

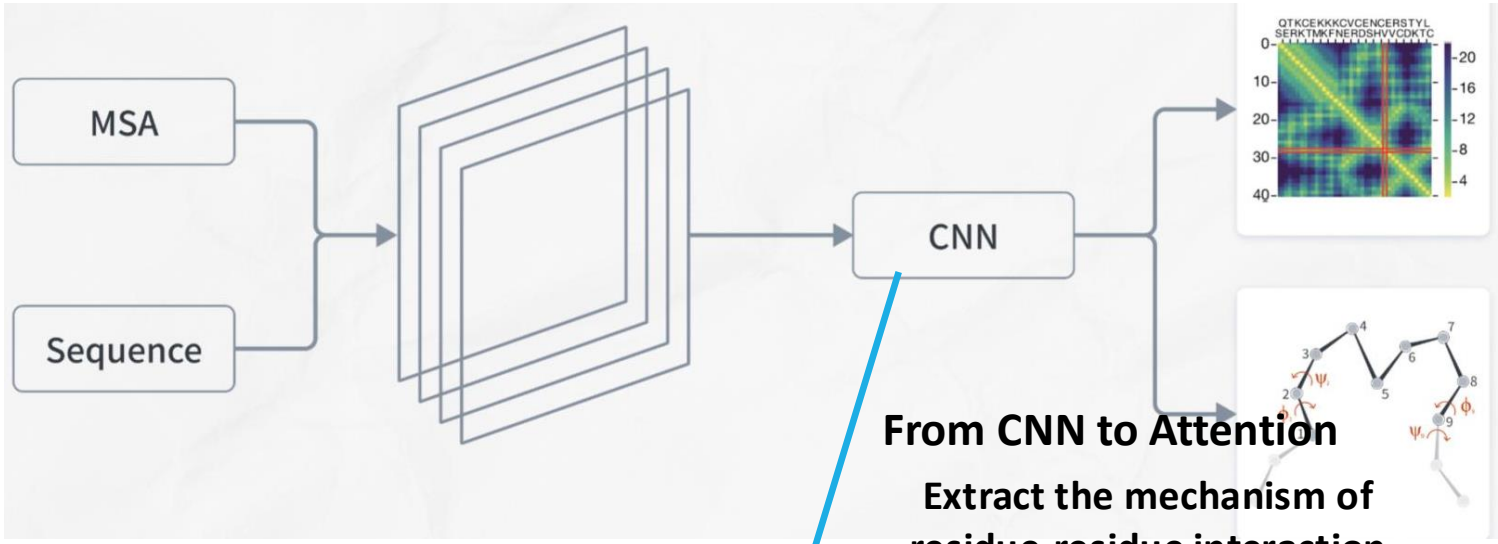
**Alphafold**

After : 75%

How about Attention-Based?

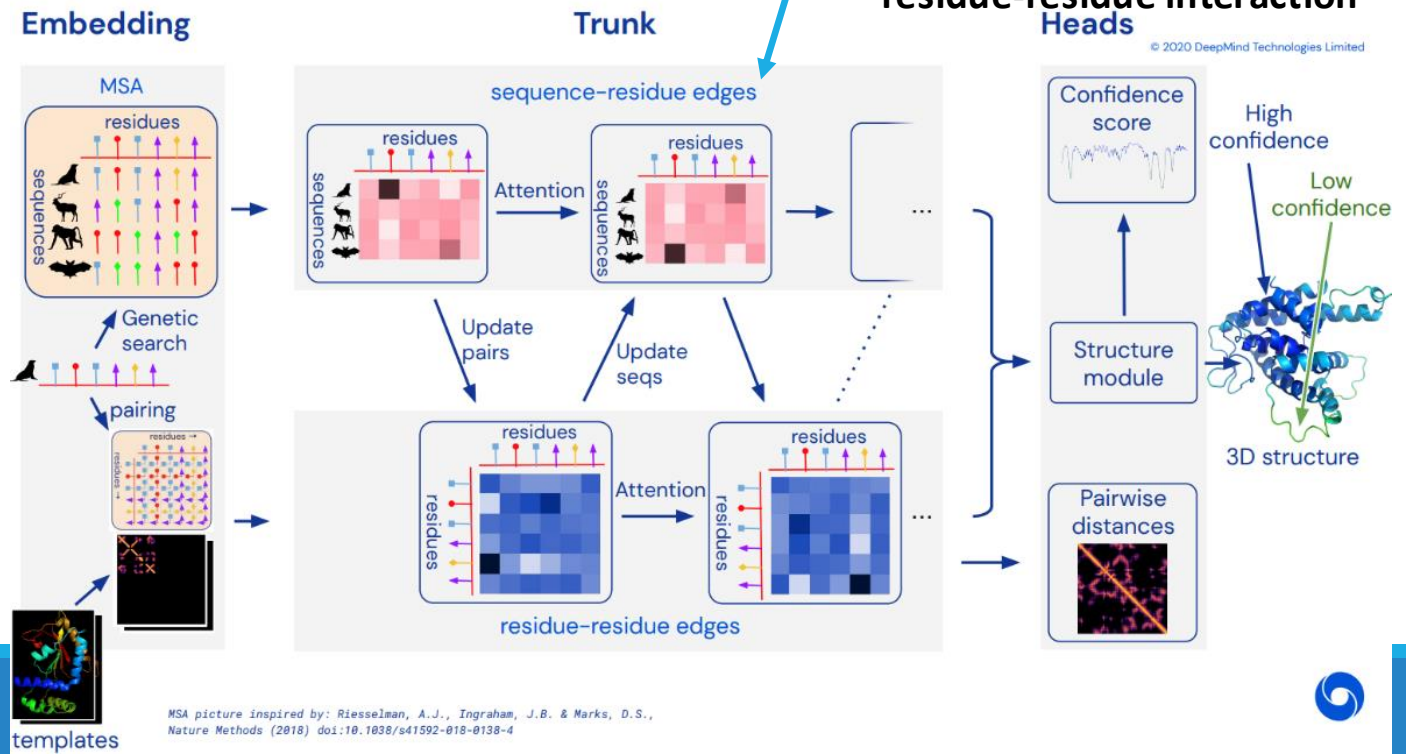
Google DeepMind

Alphafold2




**From CNN to Attention**

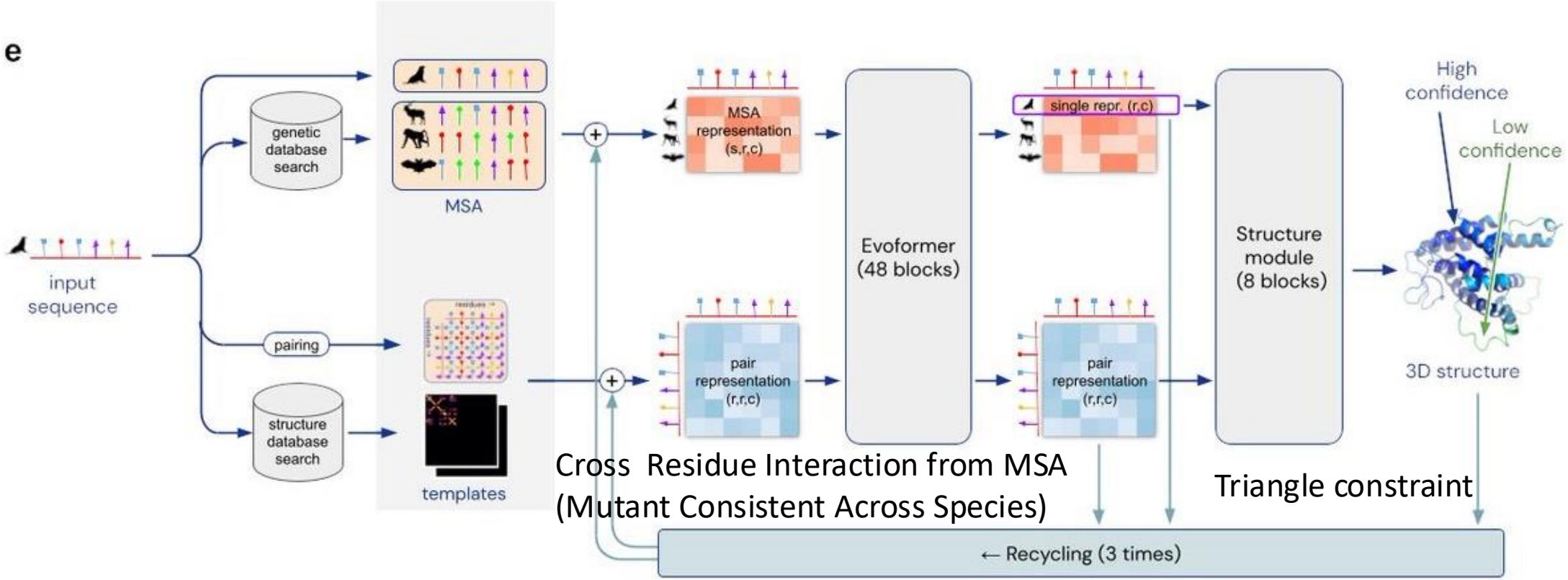
**Extract the mechanism of residue-residue interaction**




Ref. https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery

# Alphafold2





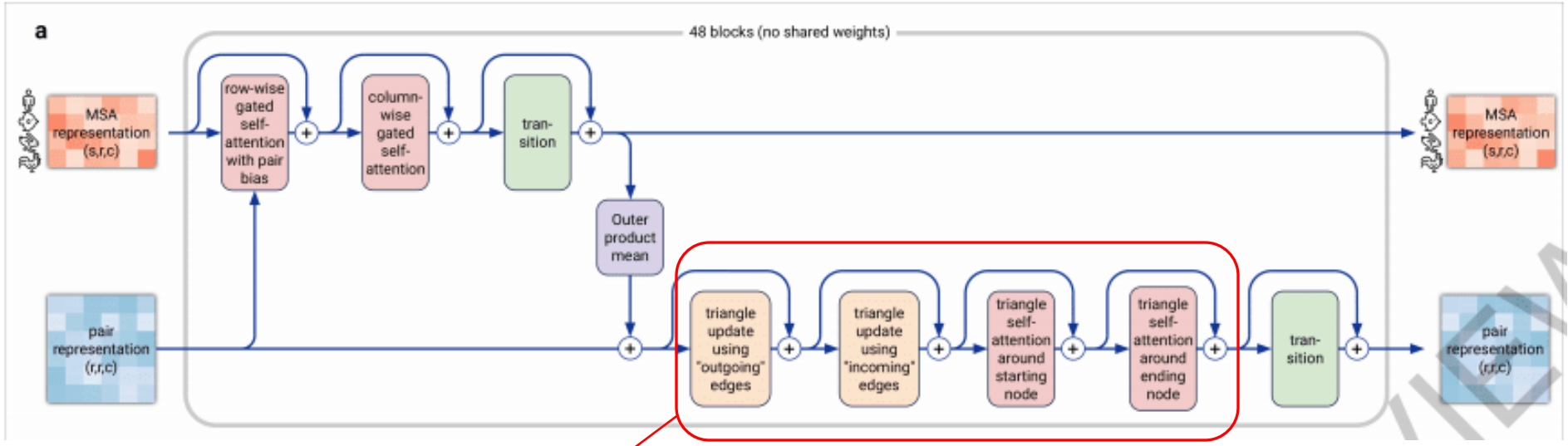

1. Training resources : 128 slides TPU v3 ≈ 300 slides GPU V100
2. Attention with Graph-Based Invariant Model Concatenate
3. Amber force Refine Side chains

Ref. https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery

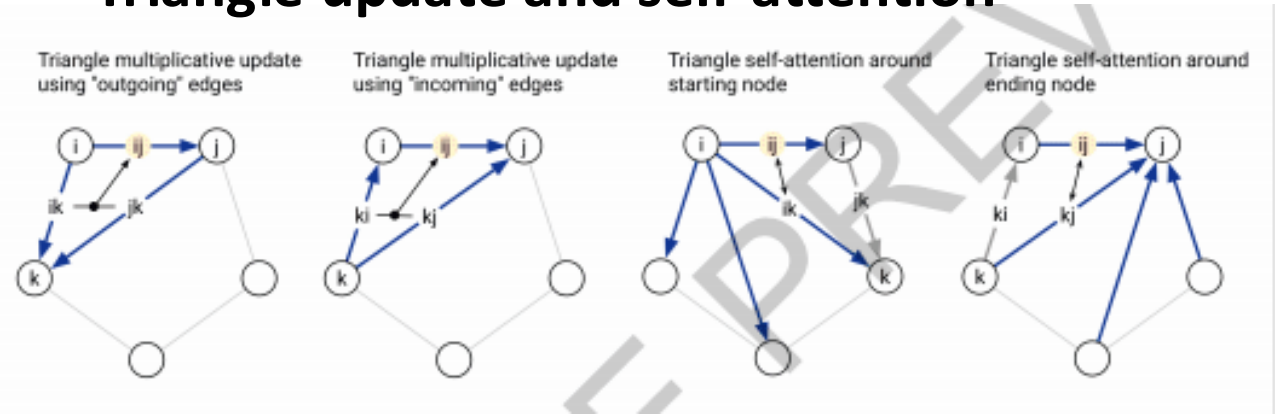
# Alphafold2



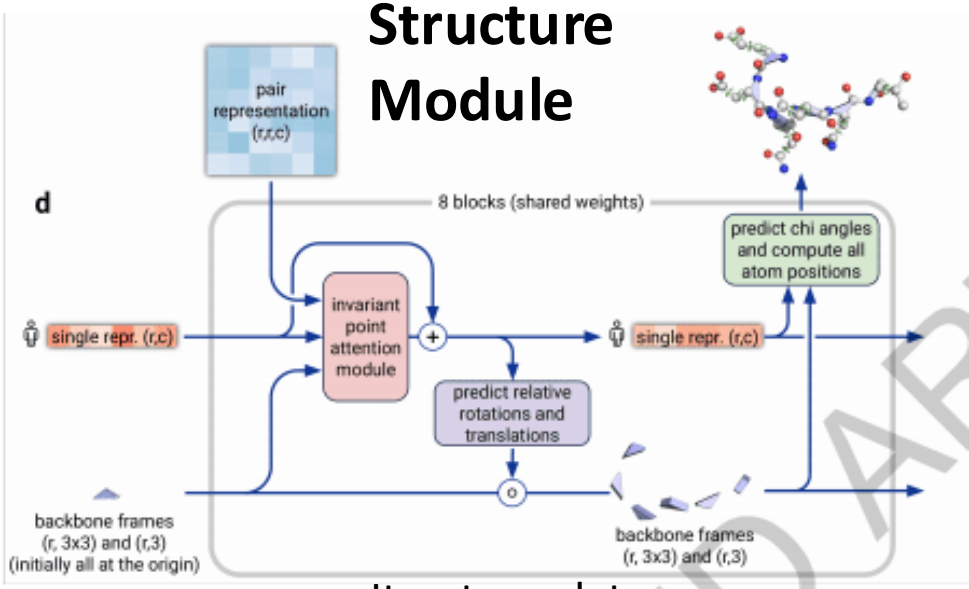
## Evoformer




## Triangle update and self-attention



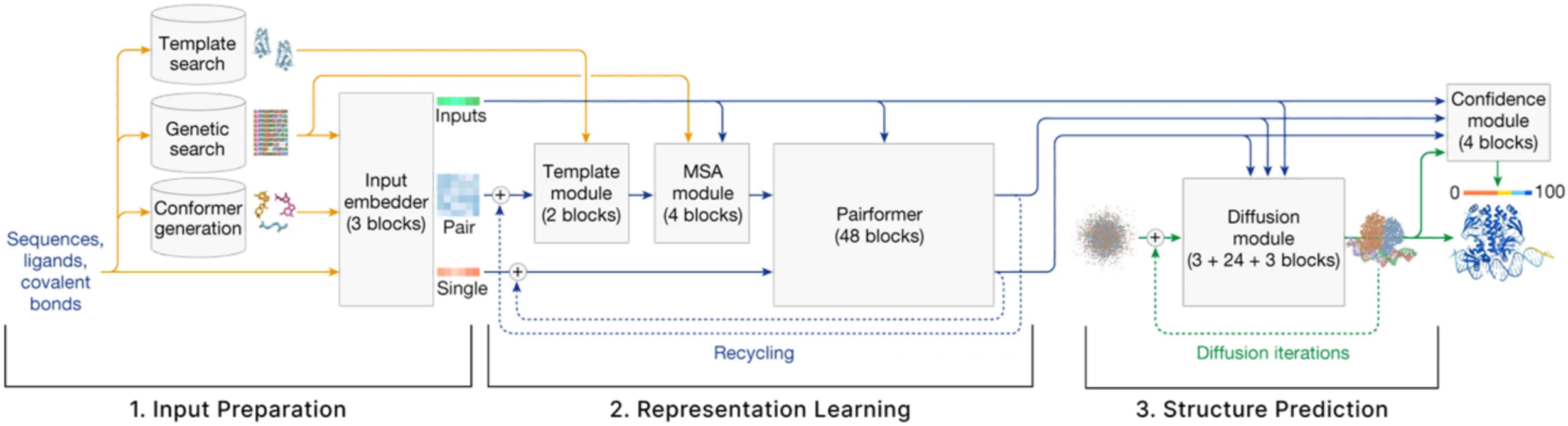

## Structure Module




Ref. https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery

Iterate update
Position And Rotation

# Alphafold3

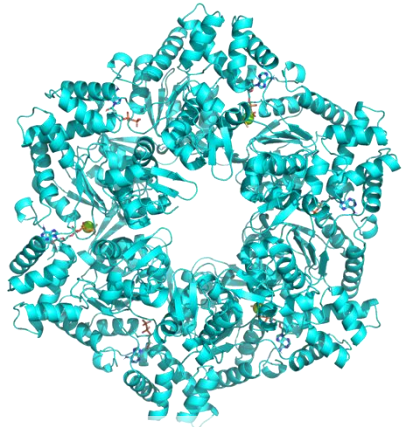



- Similar Data Structure input of Alphafold 2 – multimer.
- Module 1,2 and Alphafold2's Evoformer are very similar. Pairformer simplifies MSA computation. Put more structure computation to the mapping between structure and sequence.
- Module 3 changes Alphafold2's position-rotation iterational decoding to Denoise Diffusion-based structure reconstruction by time. Strengthen the readiness of structure.

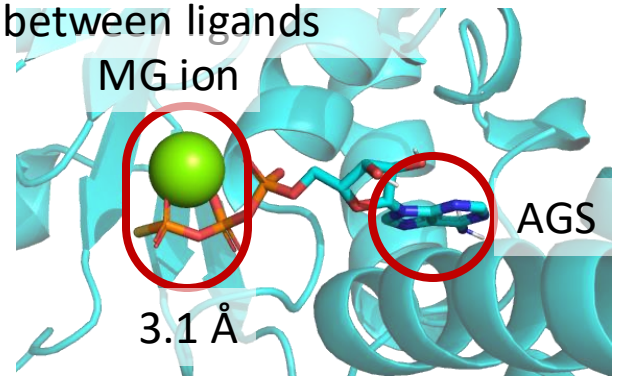
# Graphen Dynamic Structure Simulation

- Non-Covelent Force simulation via atom position and rotation by Atom Quantum Force Model
- RL-base Learning for Structure-Structure interaction dynamic simulation and accumulate Quantum Force Path

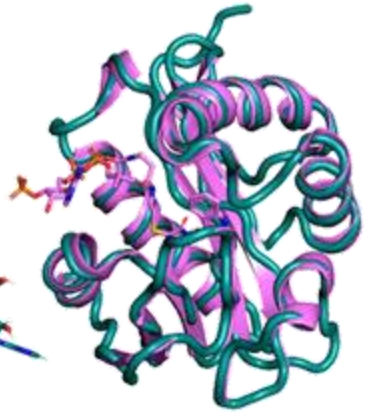

Graphen
1st : 0.77



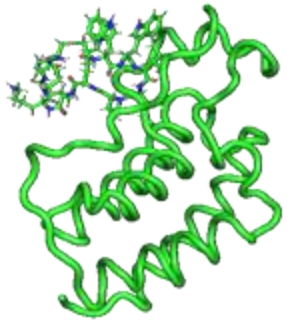
More reasonable distance between ligands




Ground True



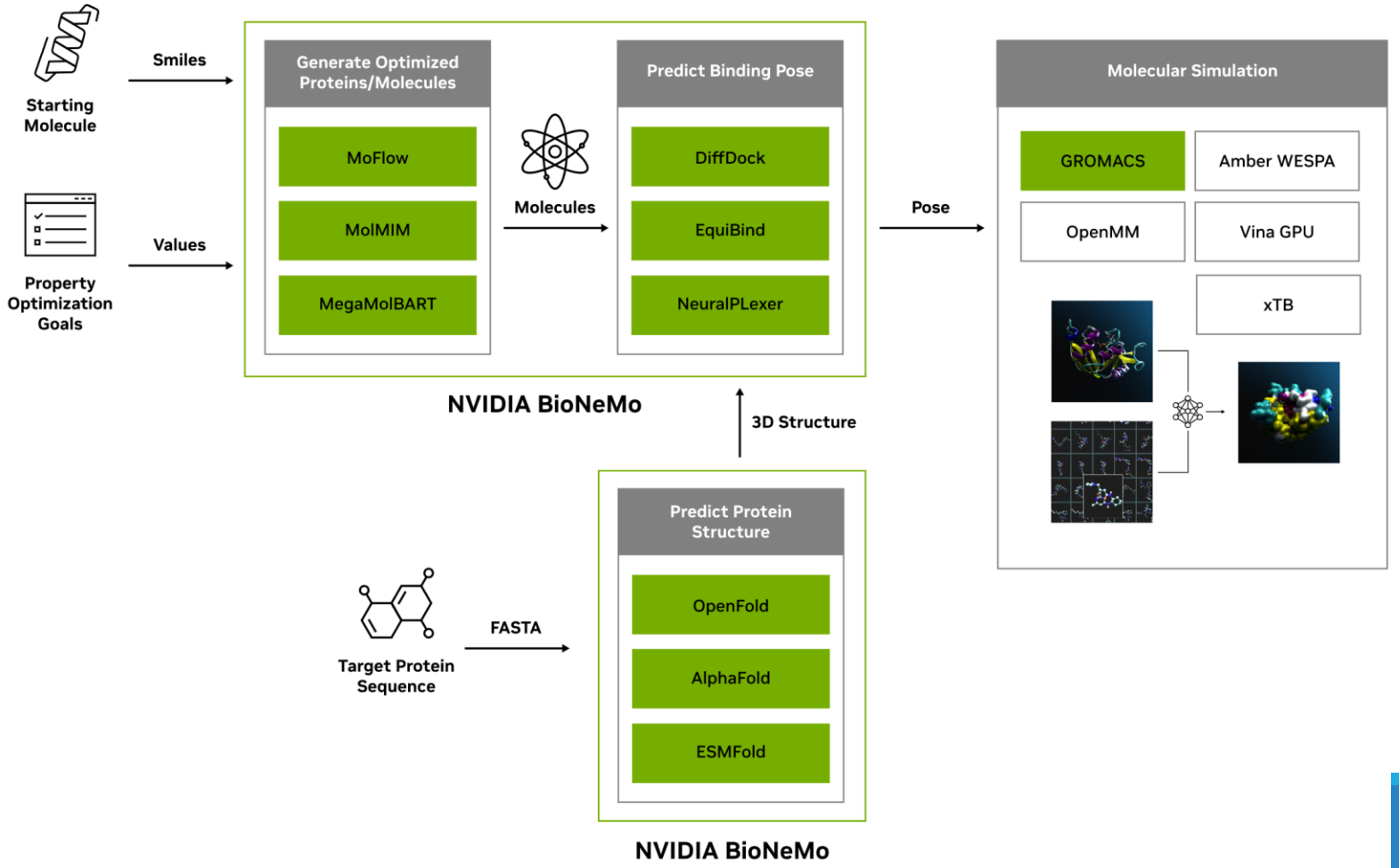
Graphen Simulation



Dynamics Interaction Change by Force estimate and Agent Simulation

# NVIDIA LLM services for drug discovery

- AlphaFold2: A deep learning model developed by DeepMind and used by over a million researchers
- DiffDock: This model predicts the 3D orientation and docking interactions of small molecules with high accuracy and computational efficiency
- MoFlow: This generative chemistry model builds molecules from scratch for molecular optimization and small molecule generation
- ProtGPT-2: This language model generates novel protein sequences, helping researchers design proteins with unique structures, properties, and functions

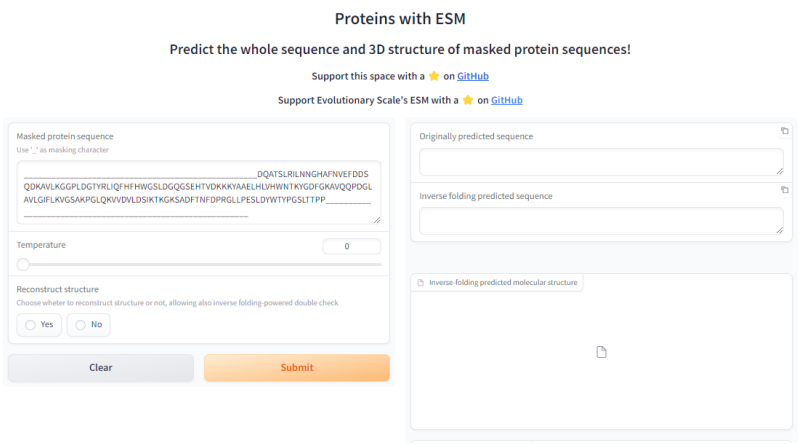




1. https://www.nvidia.com/zh-tw/clara/biopharma/
2. https://blogs.nvidia.com.tw/blog/nvidia-unveils-large-language-models-and-generative-ai-services-to-advance-life-sciences-r-d/

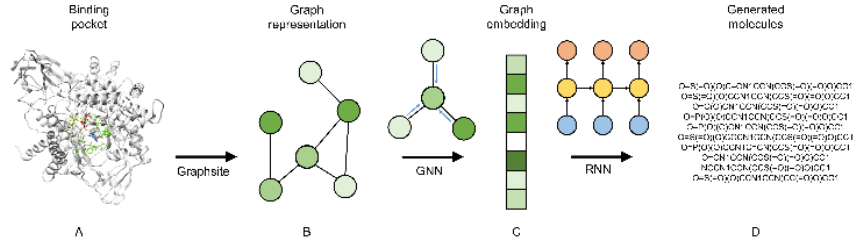
# AI in BioMedical

## Protein Design (ESM3)




https://github.com/evolutionaryscale/esm
https://huggingface.co/spaces/as-cle-bert/proteins-with-esm

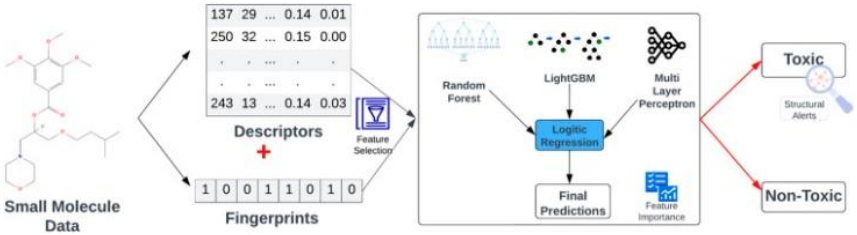
## Pocket2Drug




## Lingo3DMol

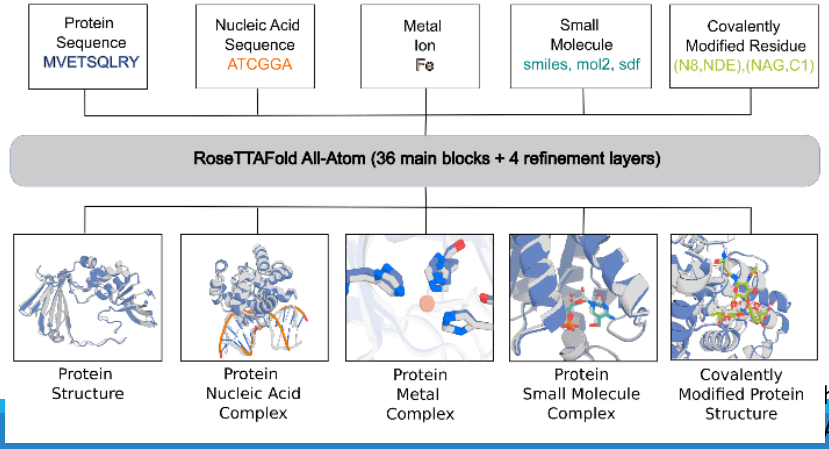
https://github.com/shiwentao00/Pocket2Drug
https://github.com/stonewiseAIDrugDesign/Lingo3DMol

## MolToxPred




Setiya A, Jani V, Sonavane U, Joshi R. MolToxPred: small molecule toxicity prediction using machine learning approach. RSC Adv. 2024 Jan 30;14(6):4201-4220. doi: 10.1039/d3ra07322j. PMID: 38292268; PMCID: PMC10826801.
https://github.com/bioinformatics-cdac/MolToxPred

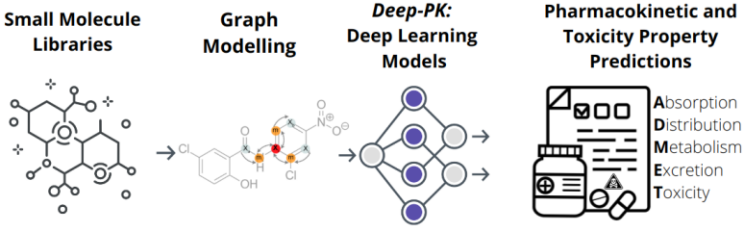
## RoseTTAFold-All-Atom




- Predicting protein structure
- Predicting protein/nucleic acid complexes
- Predicting protein/small molecule complexes
- Predicting higher-order complexes
- Predicting covalently modified proteins

https://github.com/baker-laboratory/RoseTTAFold-All-Atom?tab=readme-ov-file

## Deep-PK




https://biosig.lab.uq.edu.au/deeppk/

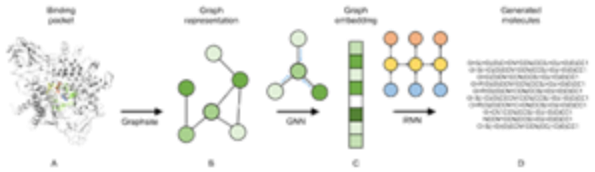
# Drug Development Procedure and Tools

Sequential tool for drug development

## Protein Design (ESM3)

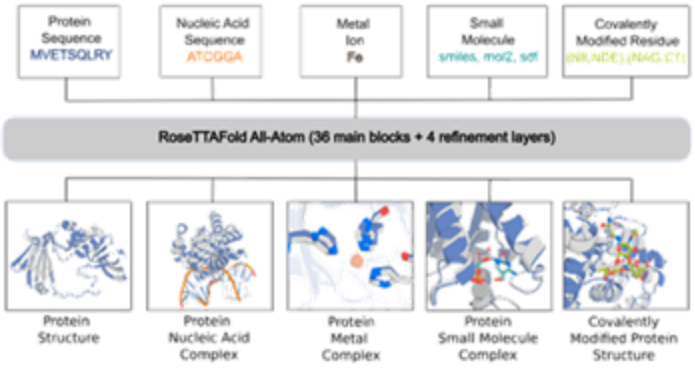
https://github.com/evolutionaryscale/esm
https://huggingface.co/spaces/as-cle-bert/proteins-with-esm

## Pocket2Drug/ Lingo3DMol



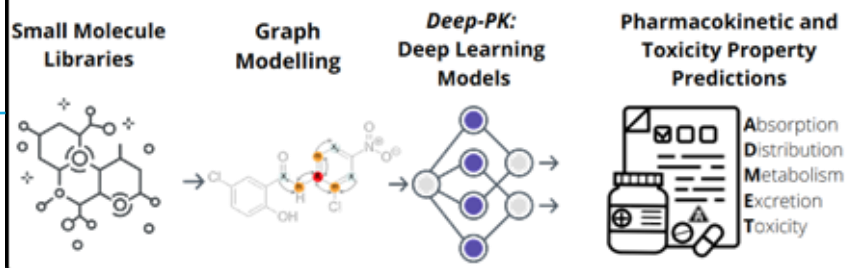
https://github.com/shiwentao00/Pocket2Drug
https://github.com/stonewiseAIDrugDesign/Lingo3DMol

## RoseTTAFold-All-Atom



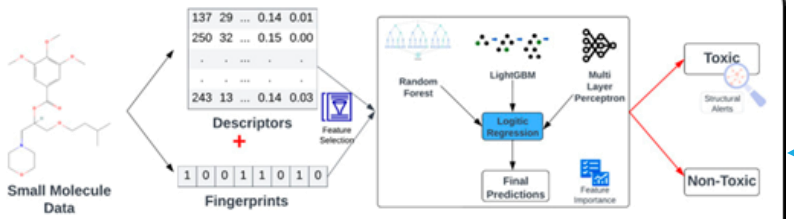

https://github.com/baker-laboratory/RoseTTAFold-All-Atom?tab=readme-ov-file

## Deep-PK




https://biosig.lab.uq.edu.au/deeppk/

## MolToxPred




Setiya A, Jani V, Sonavane U, Joshi R. MolToxPred: small molecule toxicity prediction using machine learning approach. RSC Adv. 2024 Jan 30;14(6):4201-4220. doi: 10.1039/d3ra07322j. PMID: 38292268; PMCID: PMC10826801.
https://github.com/bioinformatics-cdac/MolToxPred

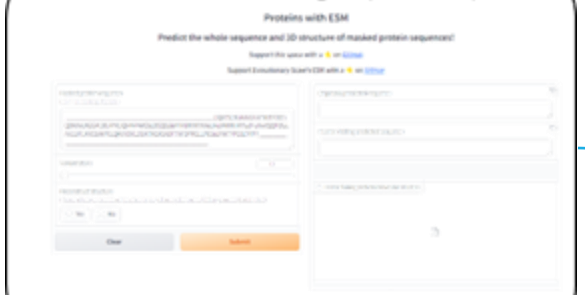
Ready For Development Or Screening

**(?)**

# Current Status and Challenges

## Sequential model Problem

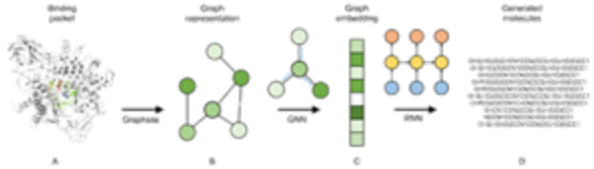
### Protein Design (ESM3)



https://github.com/evolutionaryscale/esm
https://huggingface.co/spaces/as-cle-bert/proteins-with-esm

90%

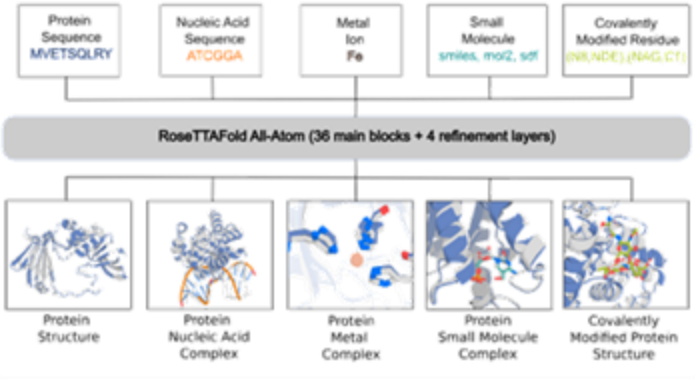
### Pocket2Drug/ Lingo3DMol



https://github.com/shiwentao00/Pocket2Drug
https://github.com/stonewiseAIDrugDesign/Lingo3DMol

90%

### RoseTTAFold-All-Atom




https://github.com/baker-laboratory/RoseTTAFold-All-Atom?tab=readme-ov-file

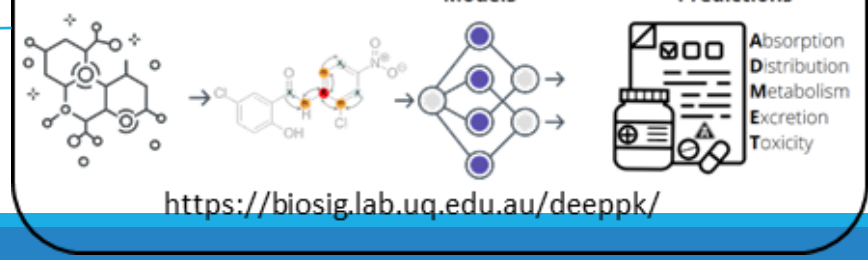
90%

**Development Looping**

- Better than the Traditional way, but still needs try-error cycle
- Not solve Error accumulation

90%

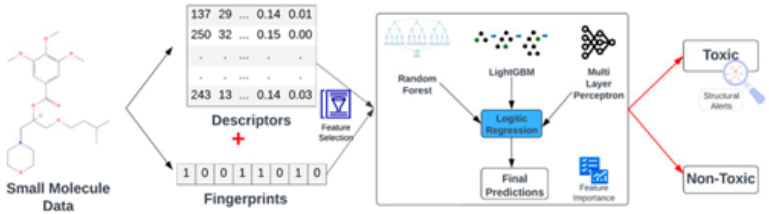
### Deep-PK




https://biosig.lab.uq.edu.au/deeppk/

90%

### MolToxPred




Setiya A, Jani V, Sonavane U, Joshi R. MolToxPred: small molecule toxicity prediction using machine learning approach. RSC Adv. 2024 Jan 30;14(6):4201-4220. doi: 10.1039/d3ra07322j. PMID: 38292268; PMCID: PMC10826801.
https://github.com/bioinformatics-cdac/MolToxPred

Exp.

**Insilico Medicine**

5,841 of 382,000 generated structure
(**1.53%** effective drug rate)

# How Graphen Solved the Problem

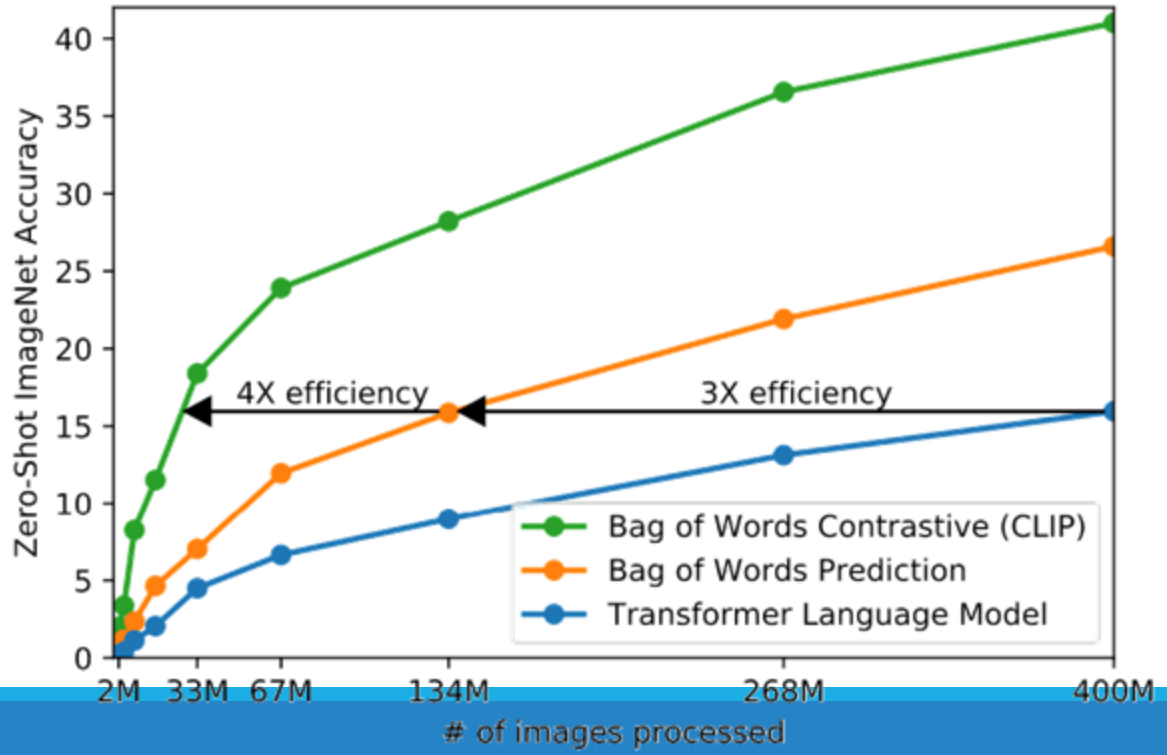
COLUMBIA UNIVERSITY

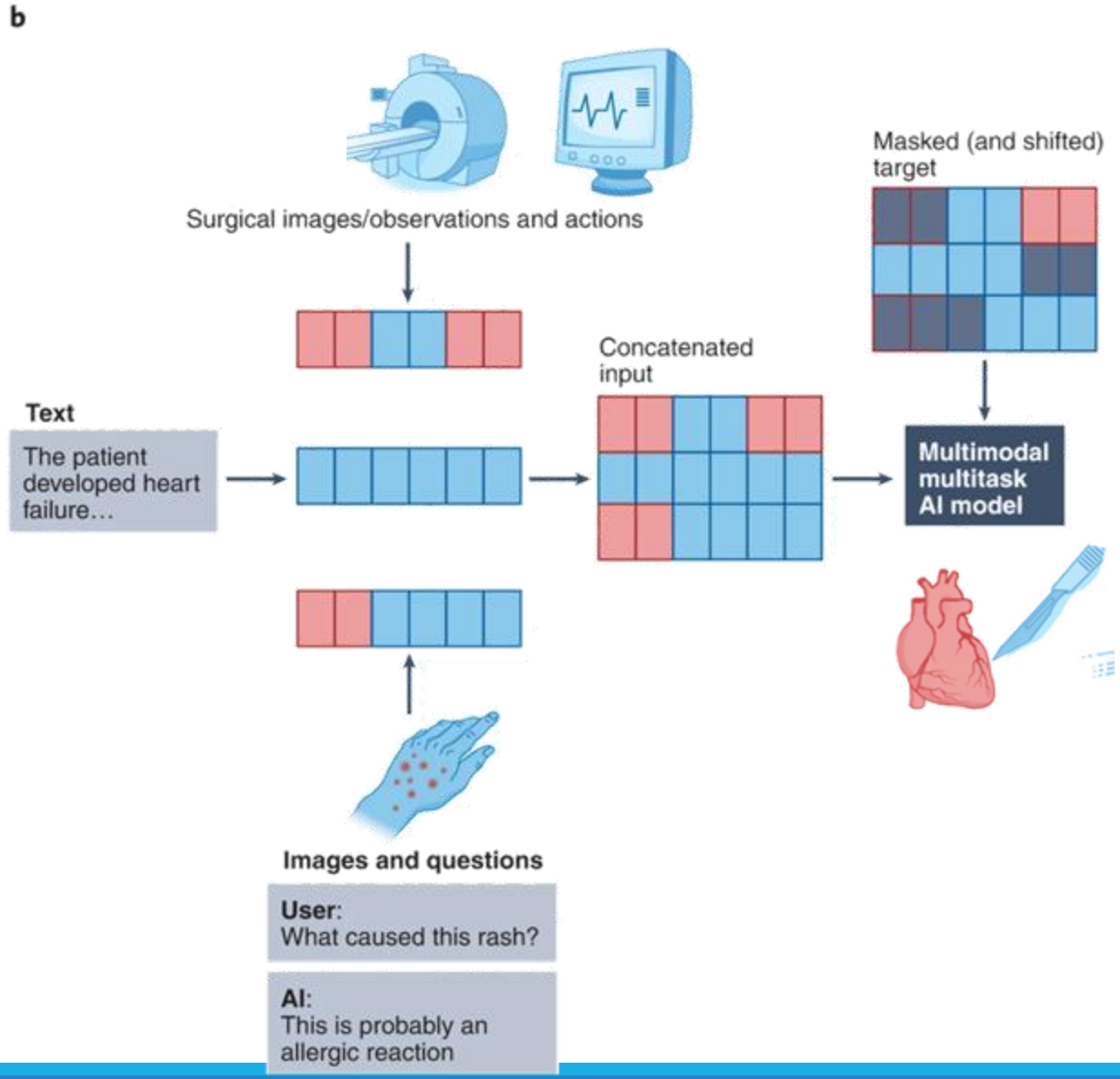
**Sequential tool for drug development** → **End to End in single step by Multi-modal**

- Reduce accumulative errors caused by sequential multiple models
- Increase the efficiency of accurate generation of new drugs
- Expand the drug features

# What is Multi-Modal

- Mixture of Features by different types of data
- Cross-relational Learning and Inference
- 4 Types of Multi-modal



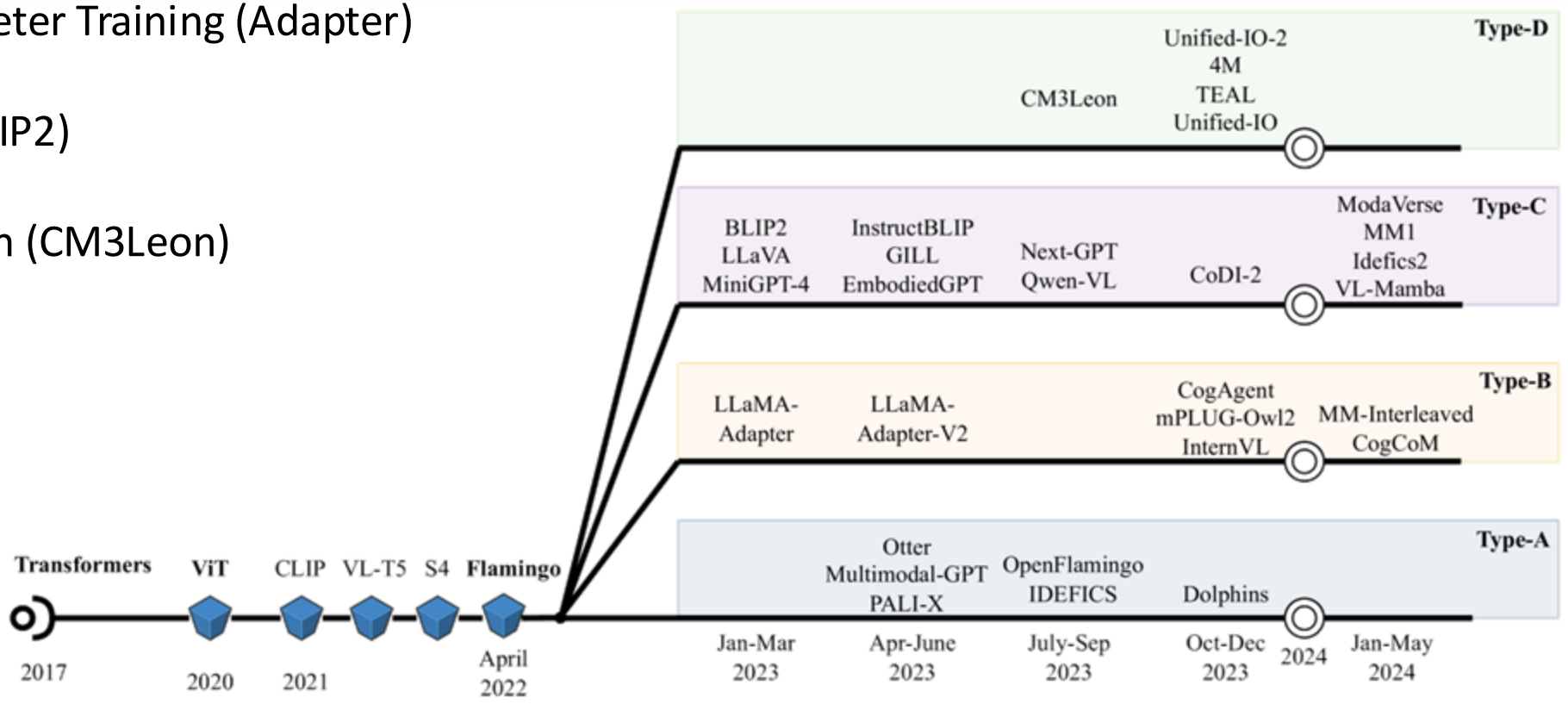
# 4 Types of Multi-Modal

Type A: High Dim. Feature Fusion (Alphafold2/3)

Type B: Additional Parameter Training (Adapter)

Type C: Cross-Training (BLIP2)

Type D: Token-base Fusion (CM3Leon)




Wadekar, S. N., Chaurasia, A., Chadha, A., & Culurciello, E. (2024). The Evolution of Multimodal Model Architectures. *arXiv preprint arXiv:2405.17927*.

# Multi-Modal in Drug Development – From Bench to Real World

Columbia University

Not Multi-modal (only string input)

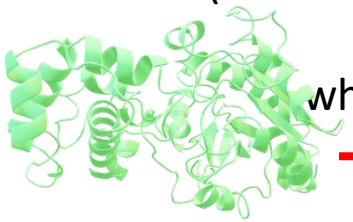
EQLLKALEFLLKELLEKL… what is this protein function

Protein Sequence | Description

Model

This is a kind of protein for Guanine nucleotide-binding.

Multi-modal (Structure + String)



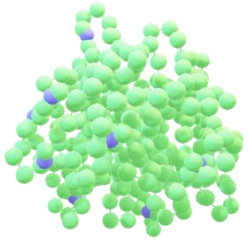
what is this protein function

Model

This is Guanine nucleotide-binding proteins function as transducers downstream of G protein-coupled receptors (GPCRs) in numerous signaling cascades，$R^{142-151}$、$R^{199-211}$、$R^{-518}$, is main function domain for GDP binding.

**Graphen Drugomics**

Target is Protein A



Oral administration

Disease is XYZ

Structure Encoder

ADME Encoder

Disease Pathway Encoder

Model

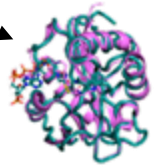
Language Decoder

SMILE: OOOOOOOOOOOOO

$IC_{50}$: < 10 nM

PK $T_{1/2}$: 12 hrs.

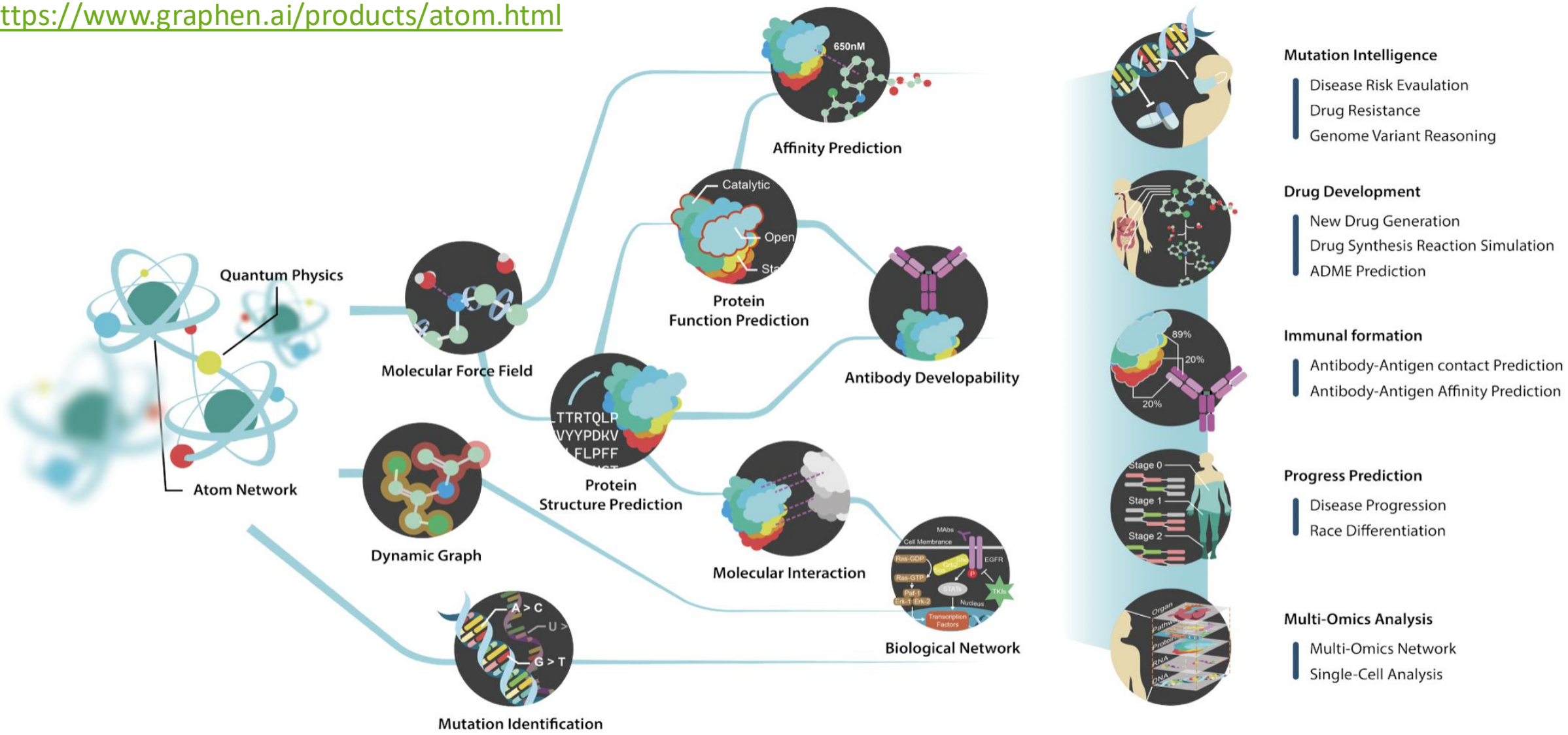
Structure Decoder



Drug-Protein A

# Graphen Atom Toolkits

https://www.graphen.ai/products/atom.html




COPYRIGHT © 2022 GRAPHEN, INC.

# Graphen Atom – a world leading AI drug platform

The world's only complete AI Drug design platform that includes 143 tools in 12 categories

https://www.graphen.ai/products/atom.html

**Atom Network**

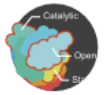
- Quantum Physics
- Dynamic Graphs

Learn More →

**Protein Structure**

- Protein Sequence
- Equivalent Graph

Learn More →



**Protein Function**

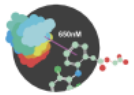
- Biological Process
- Molecular Function

Learn More →

**Molecular Interaction**

- Reaction Simulation
- Energy Score

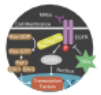
Learn More →



**Affinity Prediction**

- Energy Change
- Affinity Score

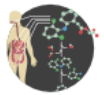
Learn More →



**Biology Networks**

- Protein Networks
- Pathway Networks

Learn More →



**ADME Prediction**

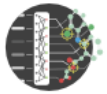
- Drug Pharmacokinetics
- ADME Score

Learn More →

**Antibody Developability**

- Antibody PDB
- TAP Score

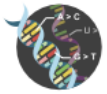
Learn More →



**Drug Generation**

- Multi-Goals
- Specific Target

Learn More →



**Mutation Intelligence**

- Virus & Human Genome
- Function Prediction

Learn More →

**Progress Prediction**

- Disease Progress
- Drug Resistence

Learn More →

**Multi-Omics Analytics**

- Cross-Omics Networks
- Single-Cell Analytics

Learn More →

# Final Project (start forming teams)

- Each team is composed of up to 3 people.
- Choose among these two areas.
  - Advanced AI Technology:
    - Multi-Modality AI
    - Perception AI
    - Expression AI
    - Reasoning AI
  - Advanced AI for Bio Science:
    - Protein-Ligan Interaction
    - RNA Structure Prediction


COPYRIGHT © PROF. C.Y. LIN, COLUMBIA UNIV.