

# An Introduction to Protein Structure Prediction with AlphaFold

Ryan Baumert, PhD

Eroglu Lab

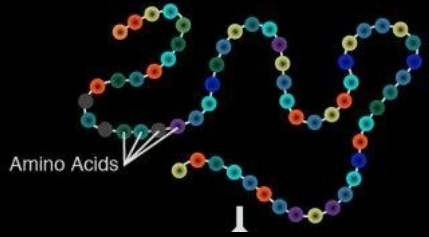
QLS

February 12<sup>th</sup>, 2025

# Seminar Overview

1. Summary of **protein structure** and modeling
2. Overview of **AlphaFold functionality**
3. **How to use** AlphaFold and why you may want to
4. Processing outputs and **next steps**

# Protein Structure Modeling – A brief history



Primary Structure



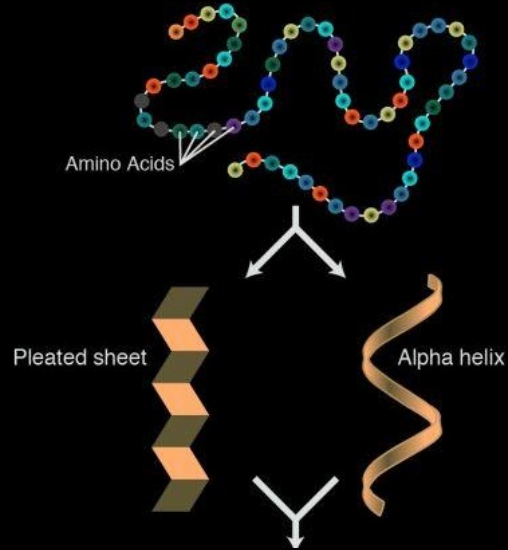
Amino acid sequence of a protein.



Christian B. Anfinsen

**Anfinsen's Dogma:** The 3-dimensional structure, under standard physiological conditions, is governed by the amino acid sequence of a protein

# Protein Structure Modeling – A brief history



Primary Structure



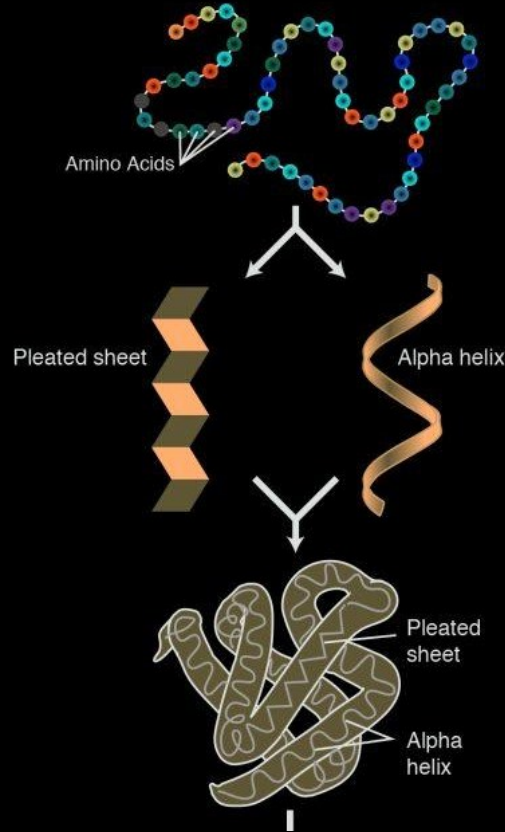
Amino acid sequence of a protein.

Secondary Structure



Hydrogen bonds cause local folding into sheets and helices

# Protein Structure Modeling – A brief history



Primary Structure

Amino acid sequence of a protein.

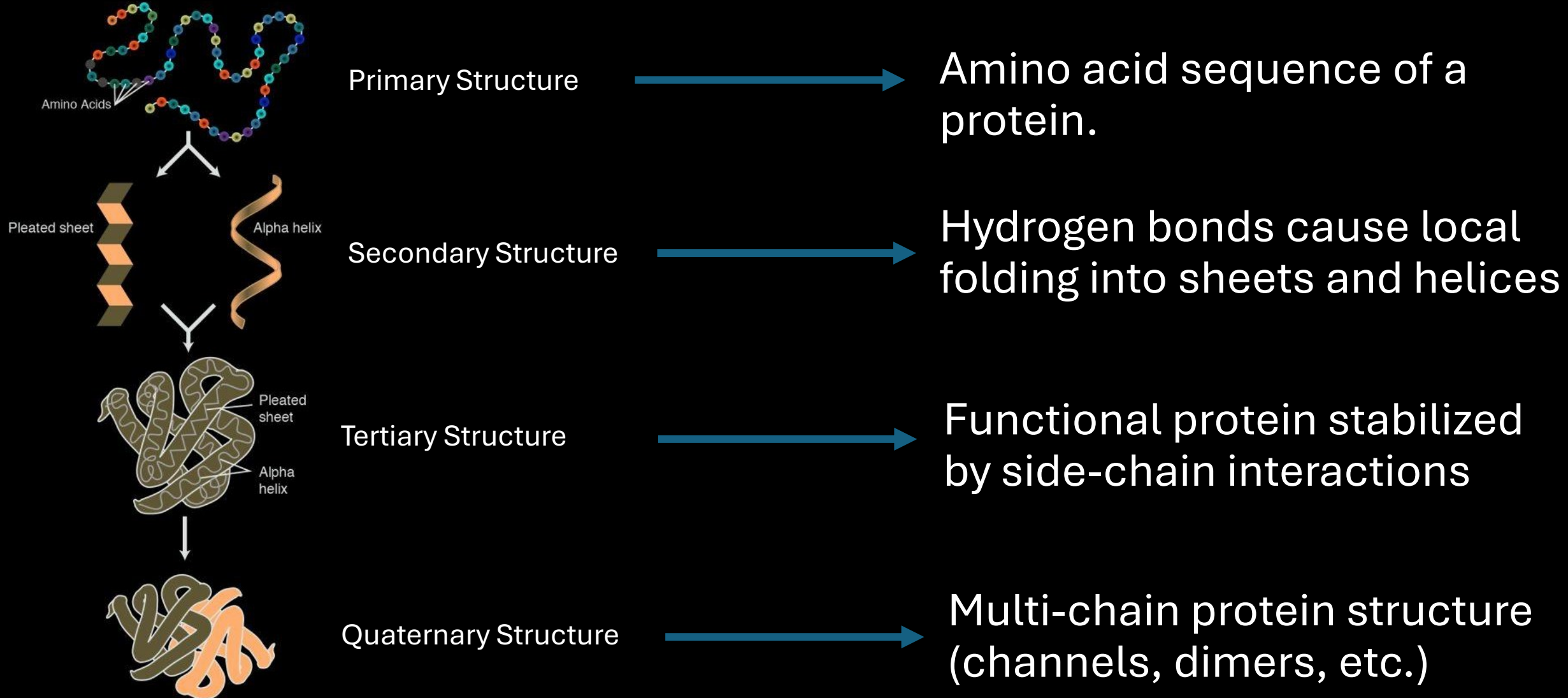
Secondary Structure

Hydrogen bonds cause local folding into sheets and helices

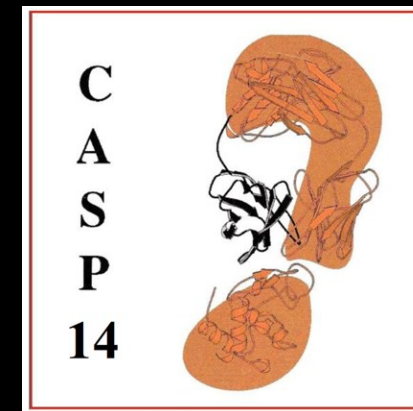
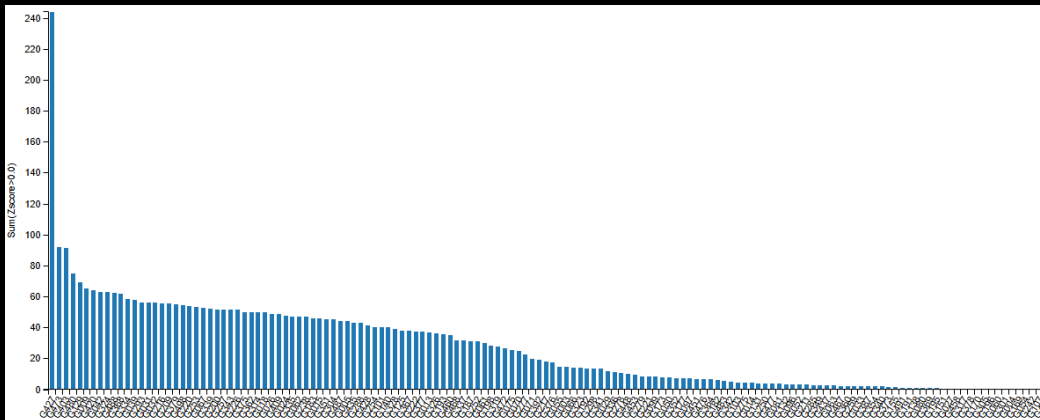
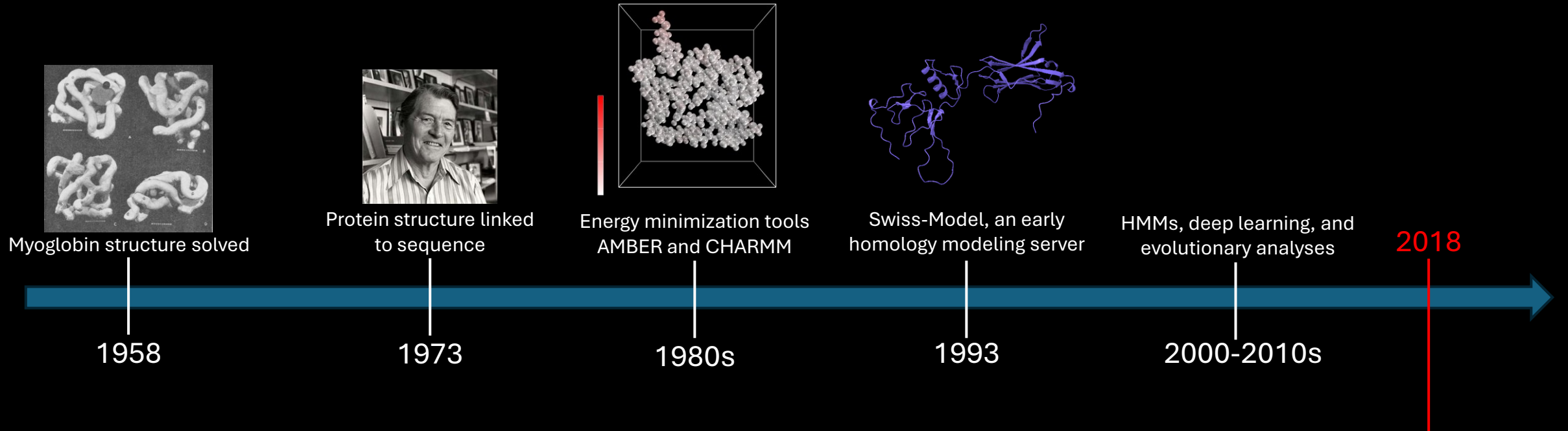
Tertiary Structure

Functional protein stabilized by side-chain interactions

# Protein Structure Modeling – A brief history

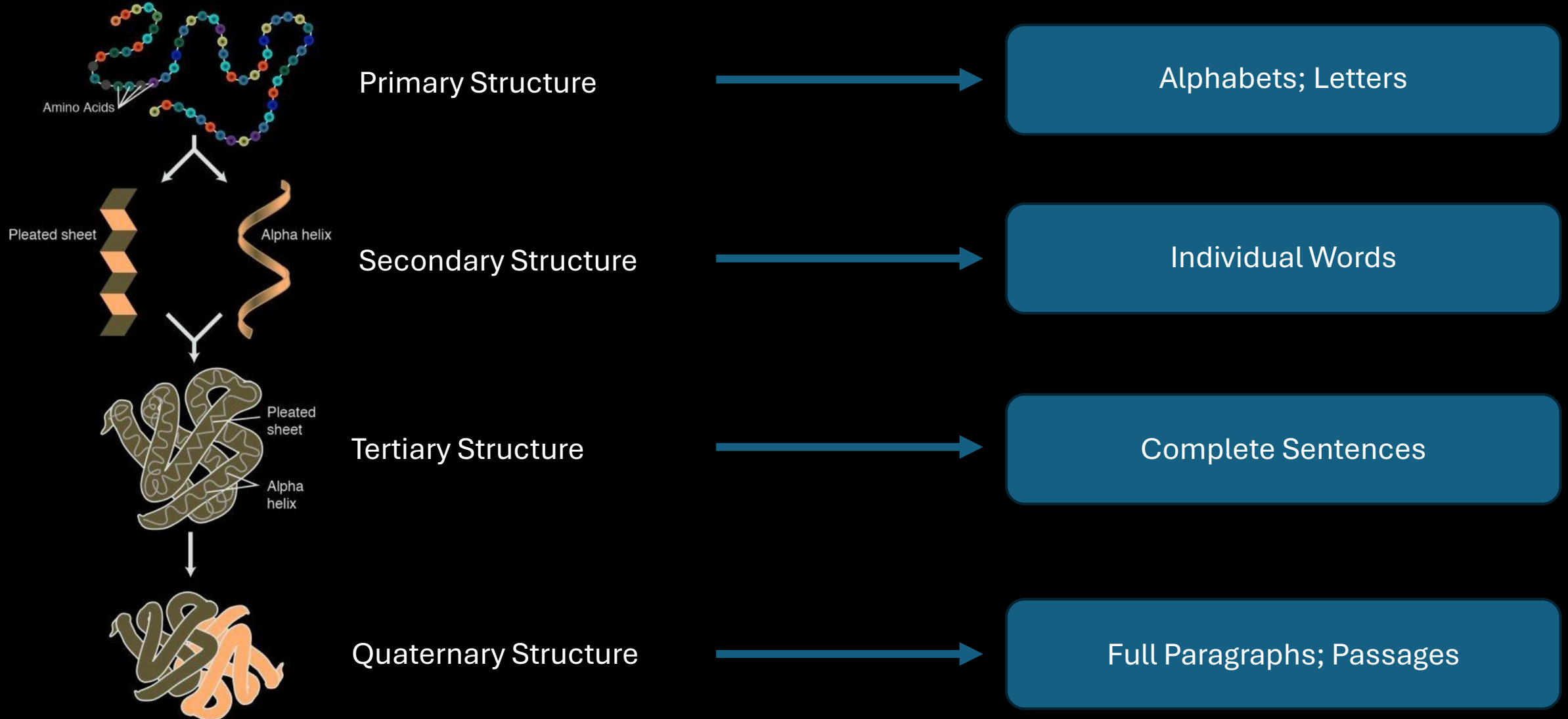


# Protein Structure Modeling – A brief history

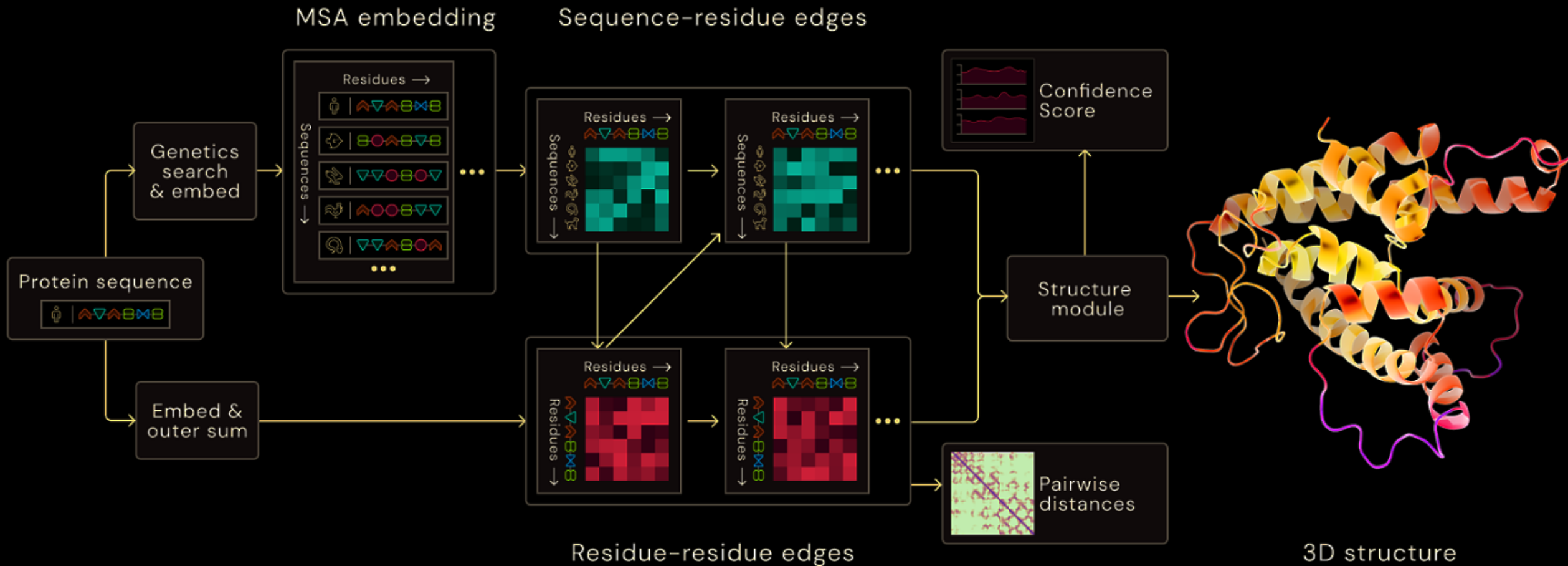


To date, Alphafold has solved the structures of over 200 million different proteins.

# AlphaFold is not a Large Language Model

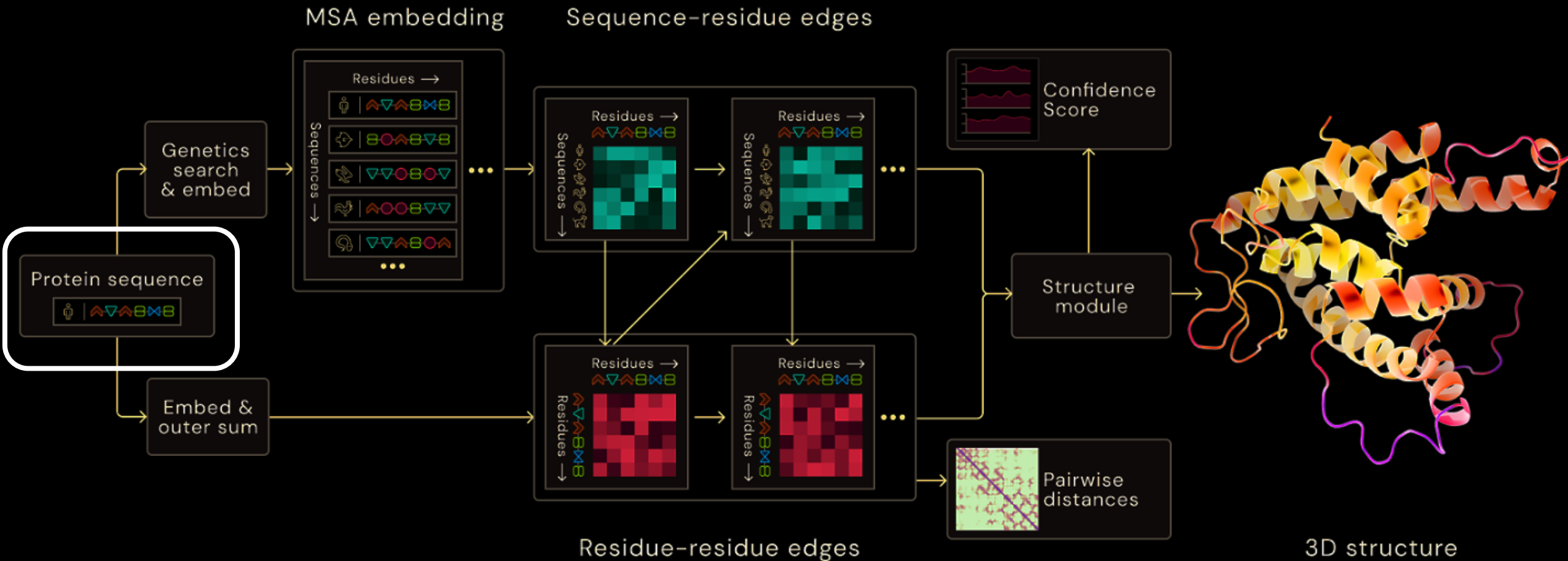


# What is AlphaFold?



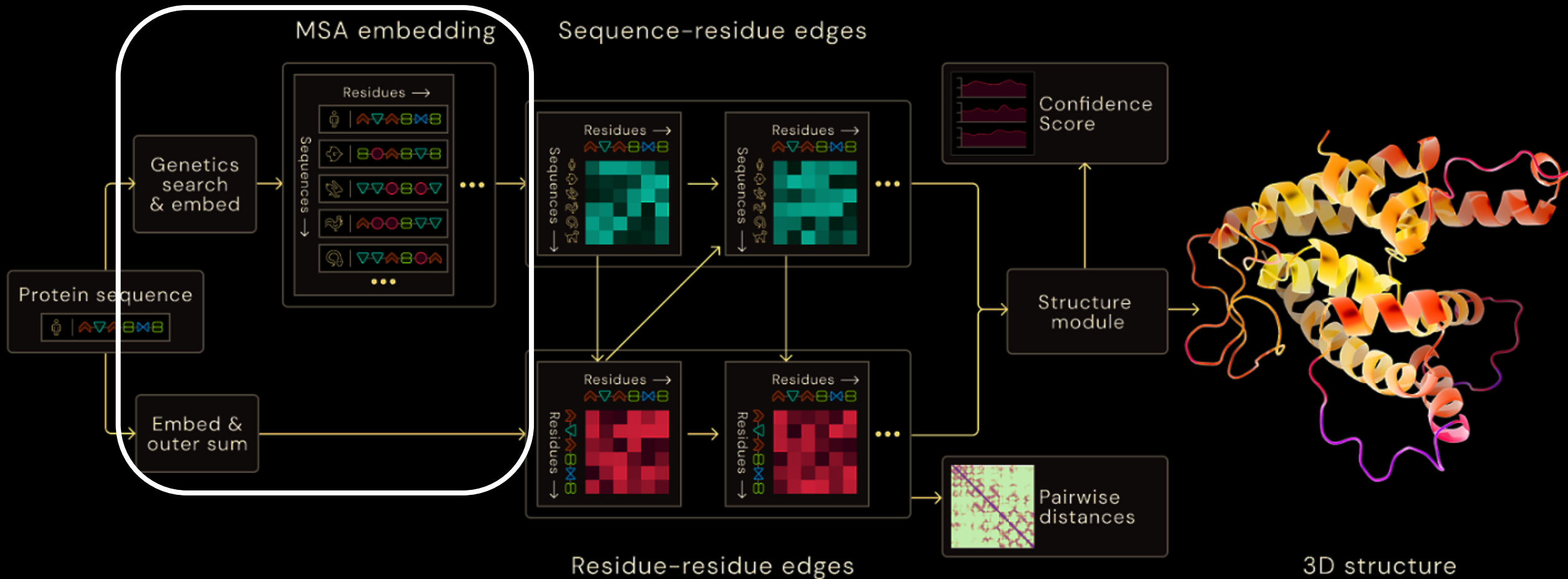
AlphaFold is a neural network-based AI model

# How does AlphaFold work? - Inputs



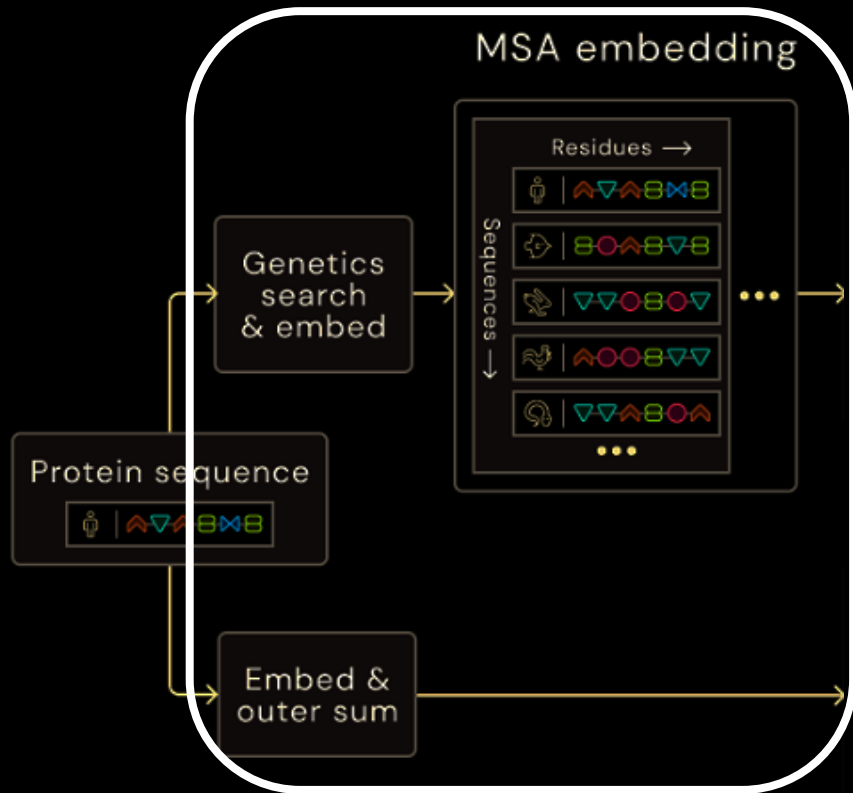
- Input protein sequence and desired cofactors (AF3: PTMs, DNA, Ions, etc)

# How does AlphaFold work? - Embedding



- Generation of sequence alignments and templates. AF is building a list of reference sequences/structures that are most similar or evolutionarily relevant to the input.

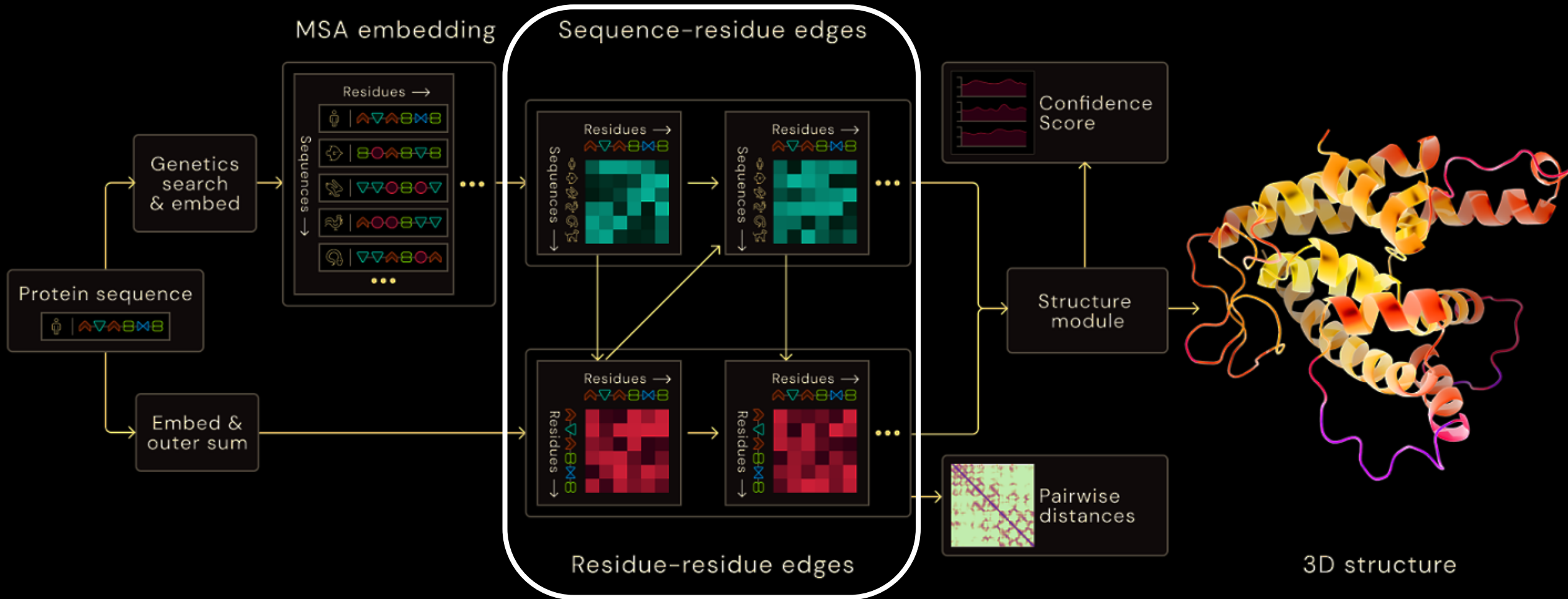
# Database search and sequence alignment



**This is where your question-dependent customizations will happen.**

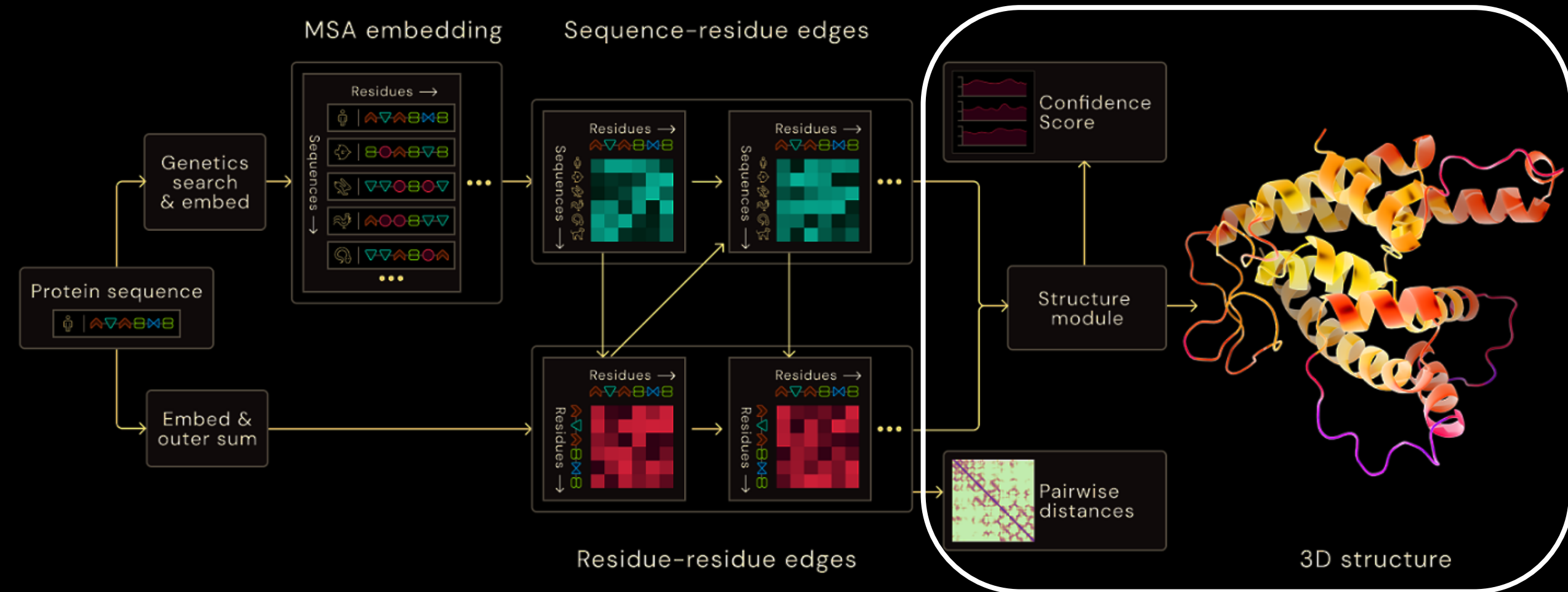
- Custom MSAs/templates can help increase accuracy of predictions in certain cases
- Mainly applicable to evolutionary queries or to proteins with very little similarity to UniREF and AlphaFold databases.
- Can be used to probe specific conformational states more accurately
- Typically, default MSAs will be best

# How does AlphaFold work? - Evoformer



- AF is generating and refining pairwise representations of residues within the protein structure and using that information to refine MSAs and repeat.

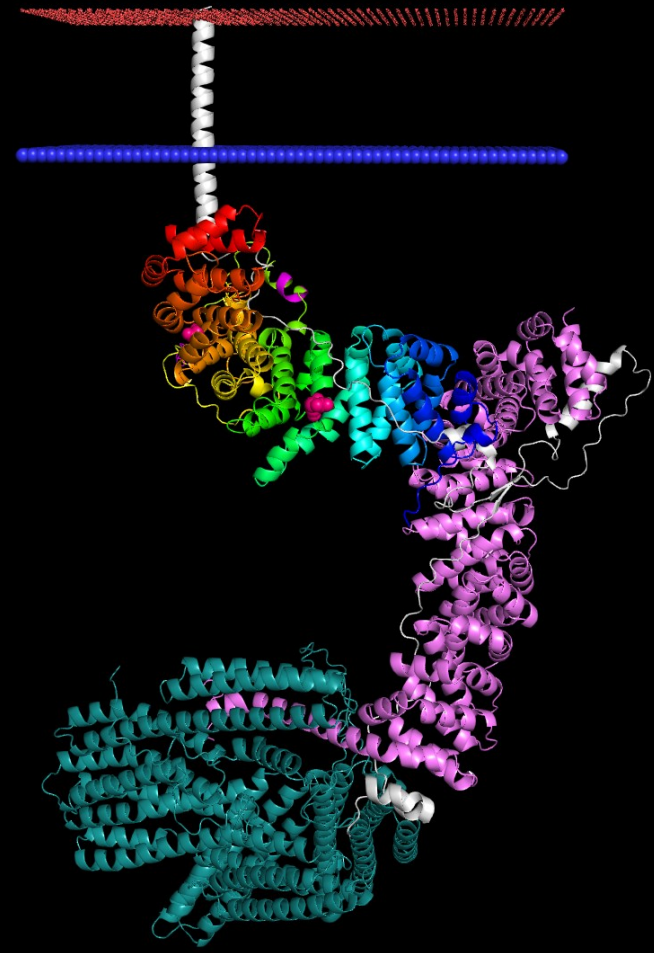
# How does AlphaFold work? – Structure Module



- AlphaFold combines the MSA and pairwise data from the Evoformer to represent amino acids in 3D space and to guide structure assembly.

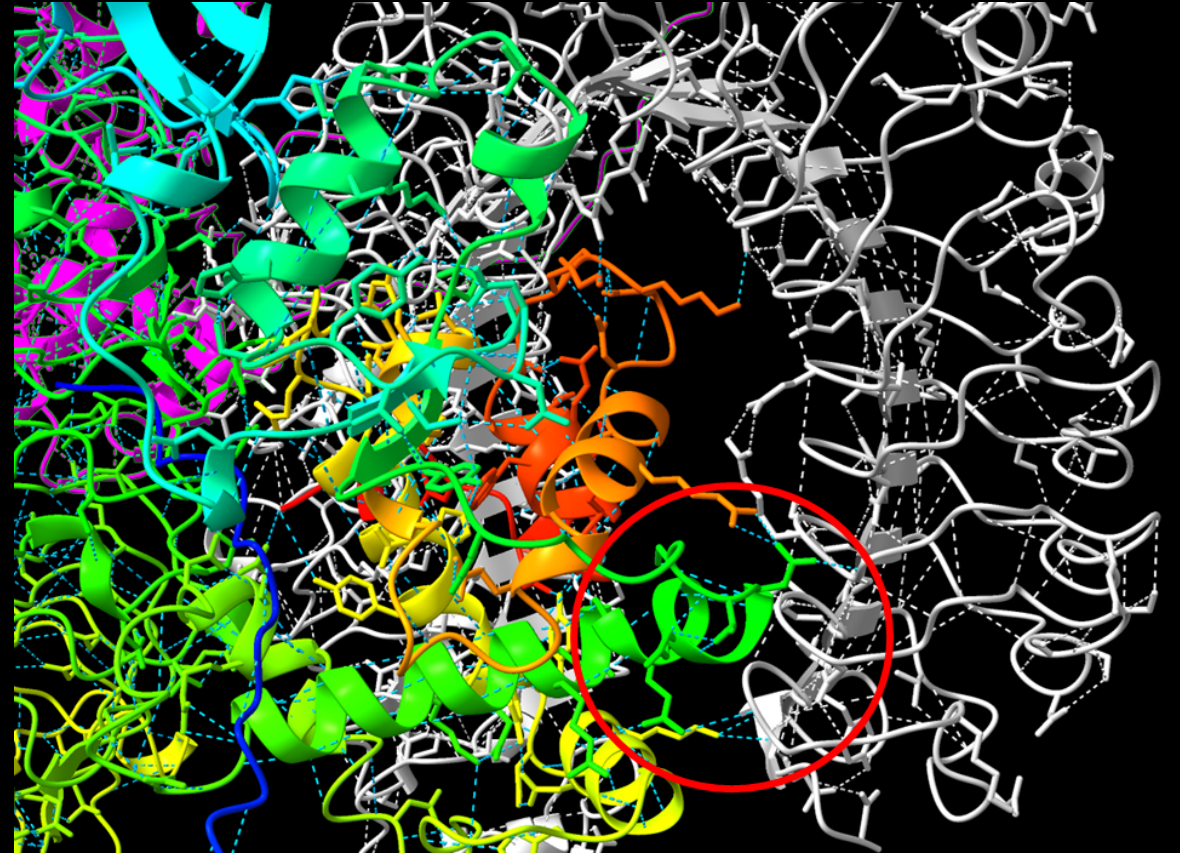
# Why should you care about protein modeling?

- Structures can add biological context to experimental observations.



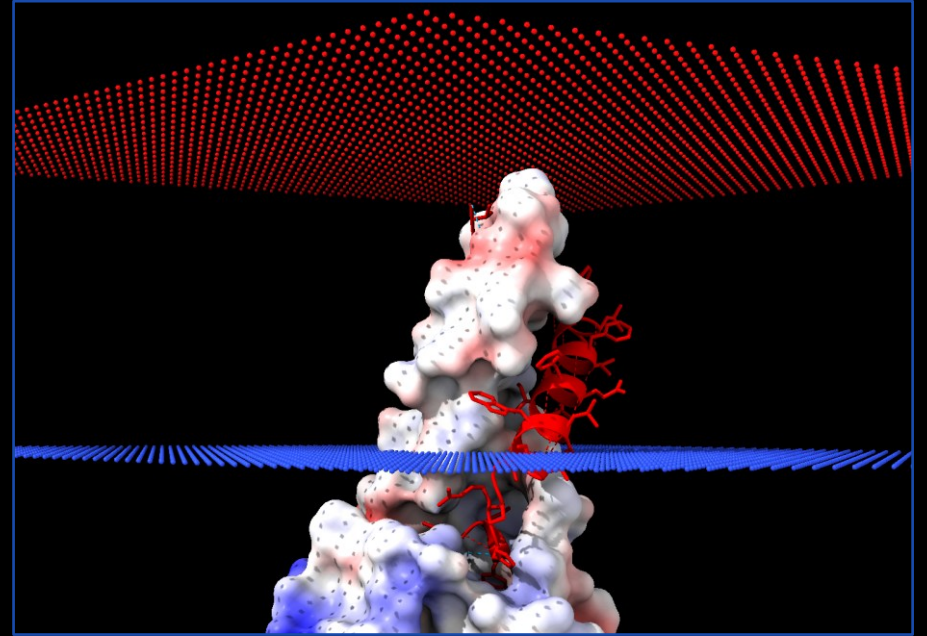
# Why should you care about protein modeling?

- Structures can add biological context to experimental observations.
- Structures can be used to guide experiment design.

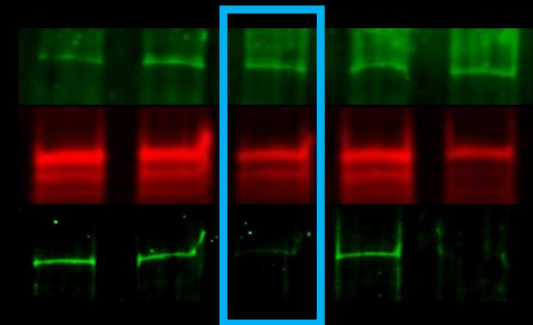


# Why should you care about protein modeling?

- Structures can add biological context to experimental observations.
- Structures can be used to guide experiment design.
- Can even be used experimentally to facilitate wet lab assays.



Predicted mutant



# Why should you care about protein modeling?

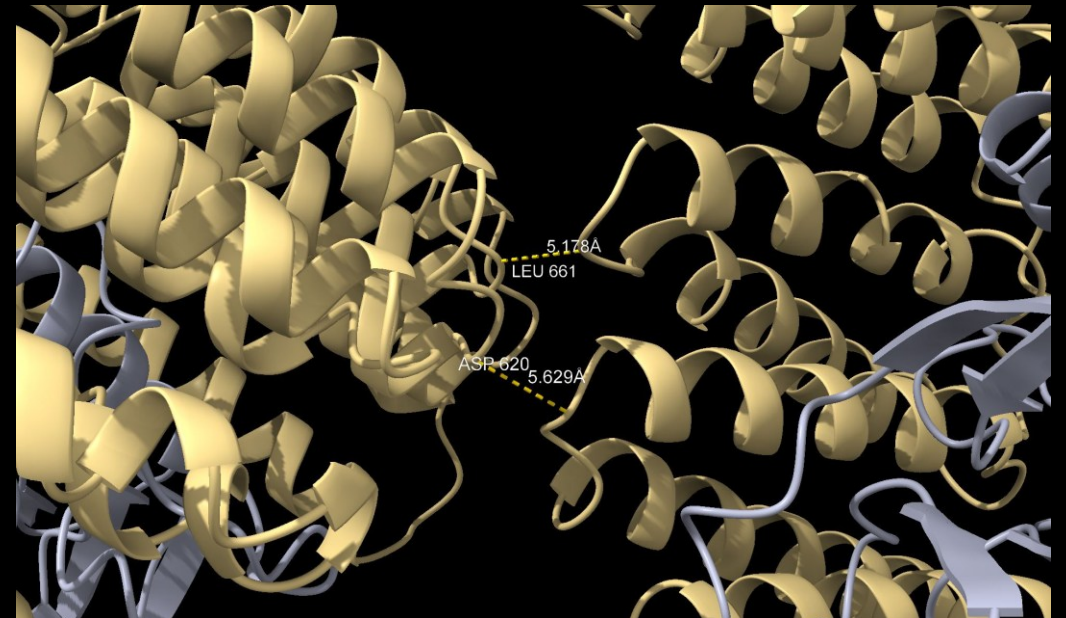
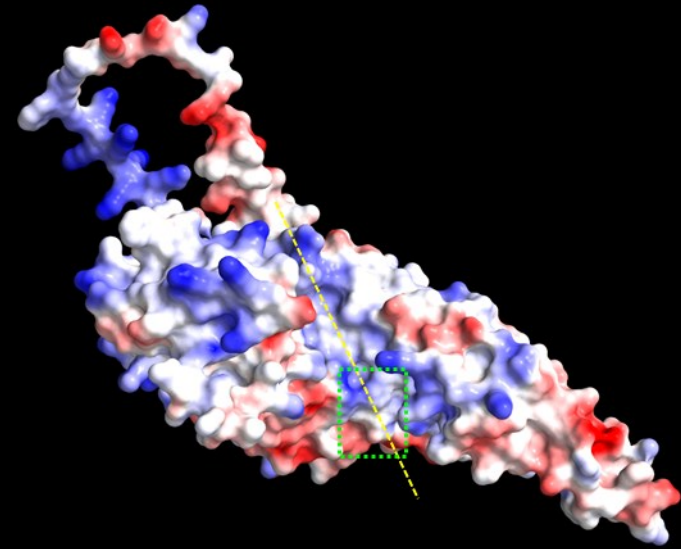
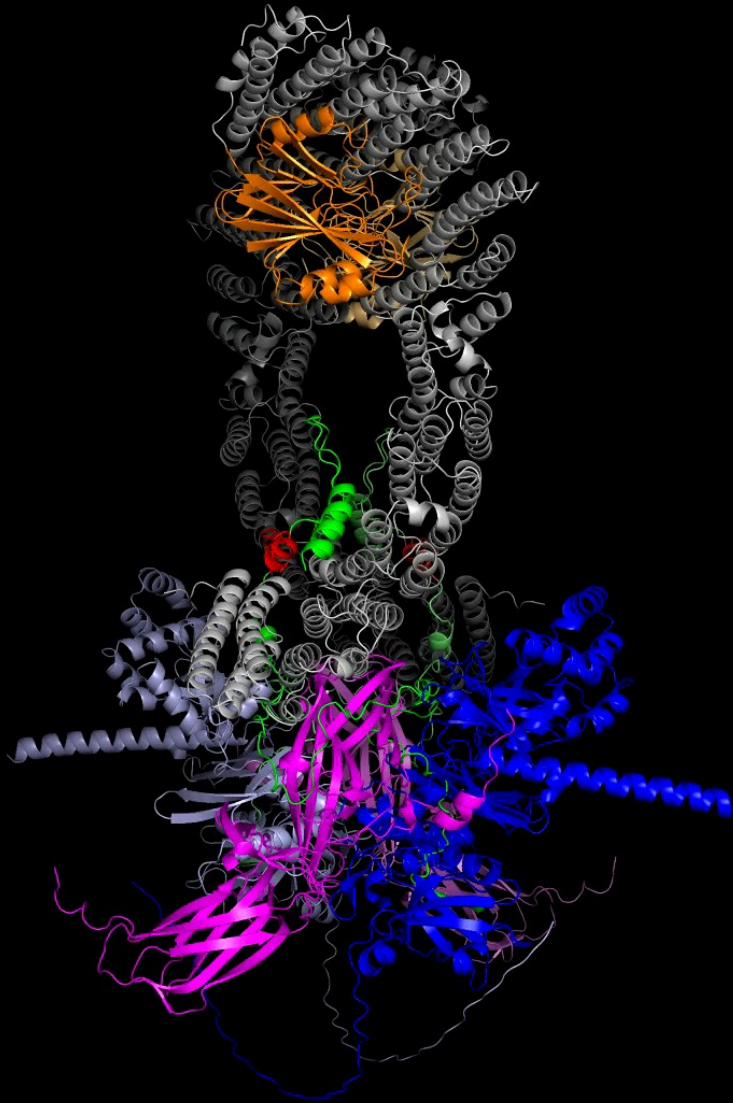
- Structures can add biological context to experimental observations.
- Structures can be used to guide experiment design.
- Can even be used experimentally to facilitate wet lab assays.

**But most of all:**

- **It is easily accessible!**

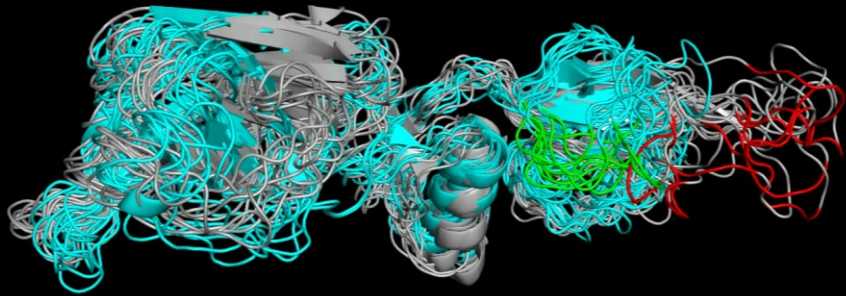


# Biological context: Assess impacts of disease-linked mutations

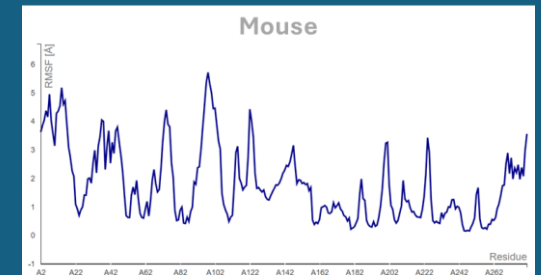
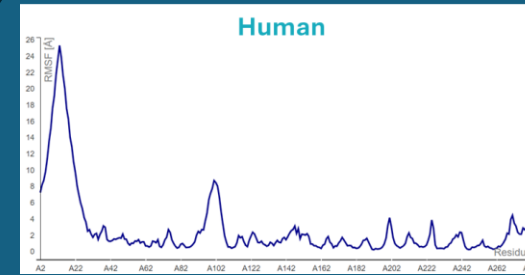


# Biological context: Identify clues of functional differences between species

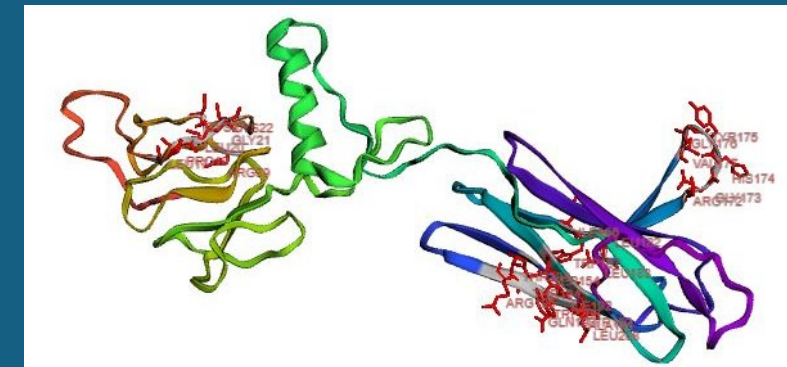
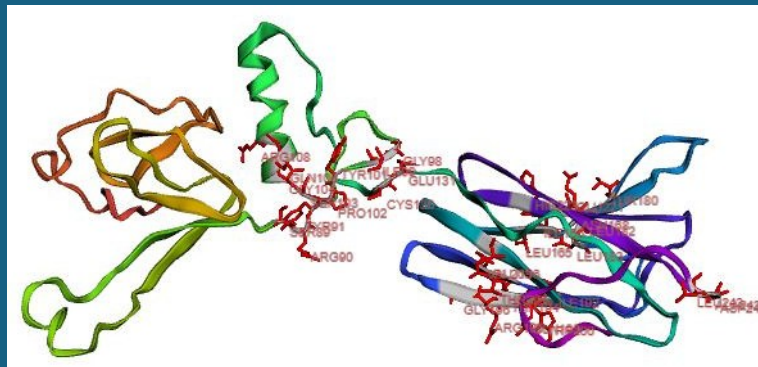
Predicted structures



Flexibility analyses

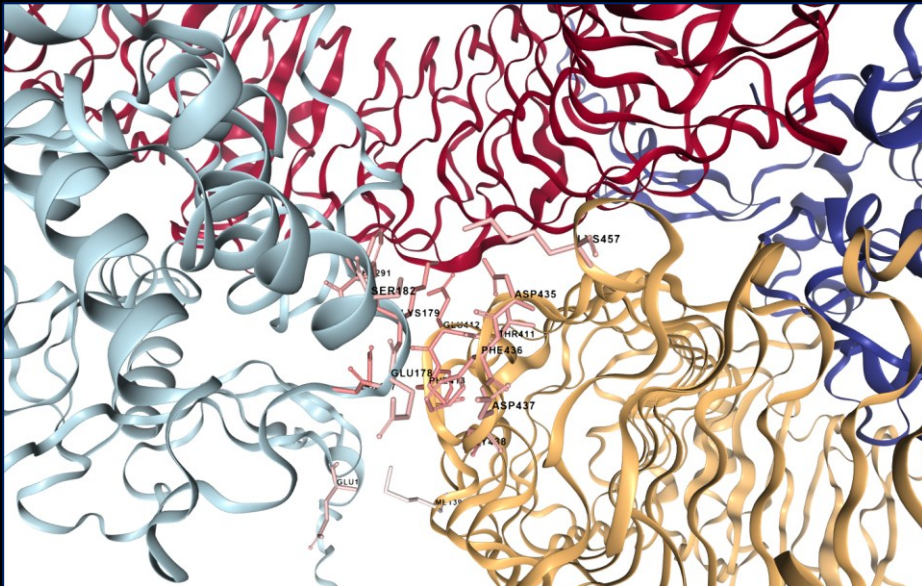


Exposed residue analyses



## Experiment Design: Identify interaction-disrupting point mutations and modes of competition

- Downstream pipelines can generate a tremendous amount of information from AlphaFold structure predictions!



# Using Alphafold

## Three Approaches:

### Local Installation

- Most involved process (requires moderate coding knowledge)
- Most customizable option
- Most accurate (in some cases)



- Unnecessary for most casual applications
- Cost prohibitive

### Colab Notebook

- Very approachable (requires minimal coding knowledge)
- Highly customizable and still benefits from dedicated GPU
- More accurate than server



- Was most common until AF3 server
- Requires subscription

### AlphaFold3 Server

- Very approachable (requires zero coding knowledge)
- Allows for PTM and non-protein modeling
- Not customizable at all



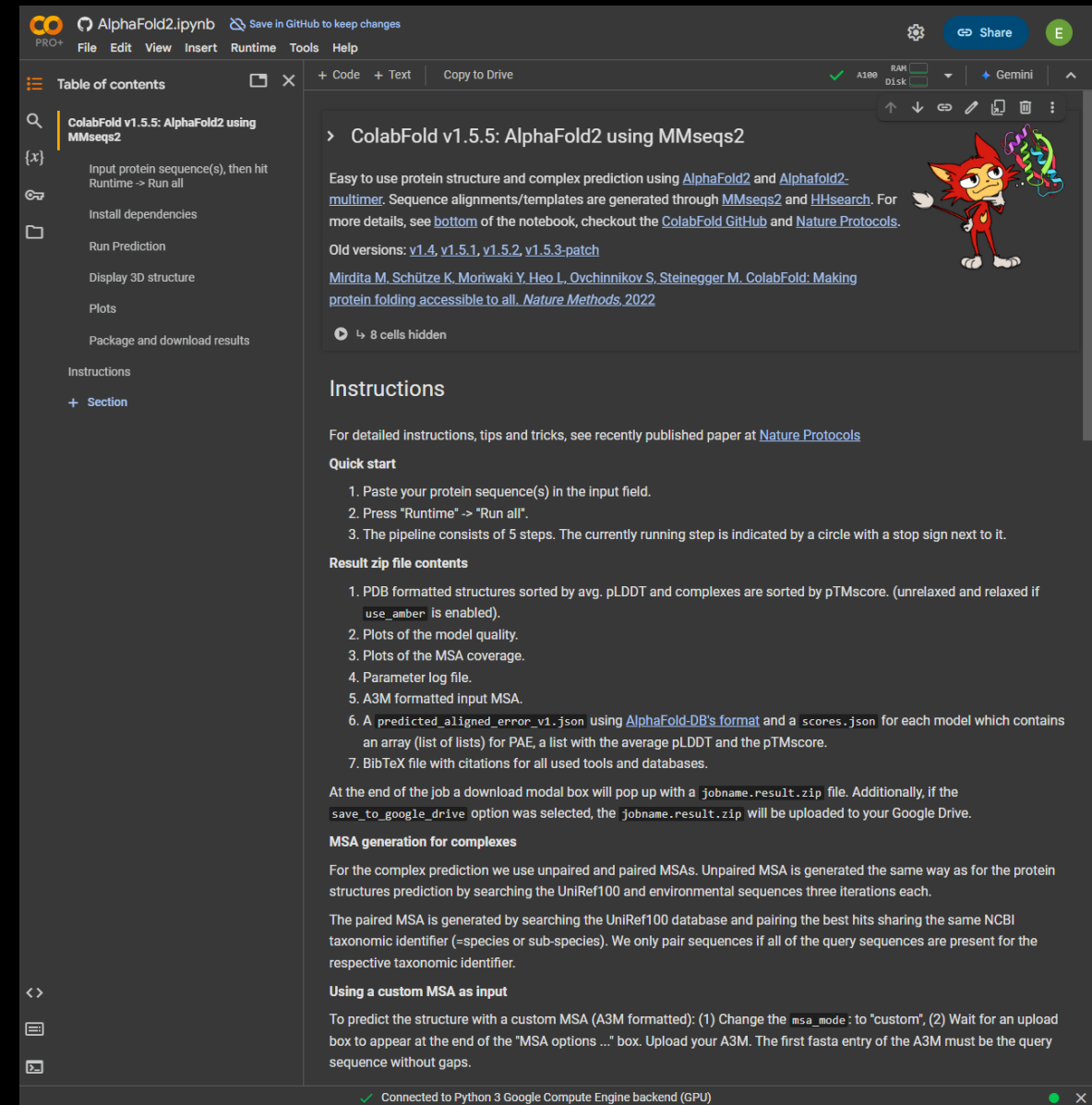
- Only way to use AF3 until recently (November 2024)
- Free

# Using Alphafold3 Locally

- Requires installation on Linux system
- Requires extensive knowledge of python language
- Requires a dedicated system
  - A100 GPU is ~ \$8,000
- I do not use AlphaFold locally.

# Using Alphafold2 ColabFold

- AlphaFold2-Multimer
  - Predecessor to AF3
  - AF3 adds non-protein inputs
- Specialized for predicting multi-chain (multi-protein) structures
- Exists as a prebuilt notebook
  - AF3 Colab requires installation



The screenshot shows the Google Colab interface for the notebook "ColabFold v1.5.5: AlphaFold2 using MMseqs2". The left sidebar contains a "Table of contents" with sections: "Input protein sequence(s), then hit Runtime -> Run all", "Install dependencies", "Run Prediction", "Display 3D structure", "Plots", "Package and download results", "Instructions", and "+ Section". The main content area is titled "ColabFold v1.5.5: AlphaFold2 using MMseqs2" and includes a cartoon fox mascot. The text describes the notebook's purpose: "Easy to use protein structure and complex prediction using AlphaFold2 and Alphafold2-multimer. Sequence alignments/templates are generated through MMseqs2 and HHsearch. For more details, see bottom of the notebook, checkout the ColabFold GitHub and Nature Protocols." It lists old versions (v1.4, v1.5.1, v1.5.2, v1.5.3-patch) and cites a paper by Mirdita M. et al. (2022). A "Quick start" section lists three steps: 1. Paste your protein sequence(s) in the input field. 2. Press "Runtime" -> "Run all". 3. The pipeline consists of 5 steps. The "Result zip file contents" section lists seven items: 1. PDB formatted structures sorted by avg. pLDDT and complexes are sorted by pTMScore. 2. Plots of the model quality. 3. Plots of the MSA coverage. 4. Parameter log file. 5. A3M formatted input MSA. 6. A predicted\_aligned\_error\_v1.json using AlphaFold-DB's format and a scores.json for each model which contains an array (list of lists) for PAE, a list with the average pLDDT and the pTMScore. 7. BibTeX file with citations for all used tools and databases. A note states: "At the end of the job a download modal box will pop up with a jobname.result.zip file. Additionally, if the save\_to\_google\_drive option was selected, the jobname.result.zip will be uploaded to your Google Drive." The "MSA generation for complexes" section explains the use of unpaired and paired MSAs. The "Using a custom MSA as input" section provides instructions for using a custom A3M MSA. The bottom status bar indicates "Connected to Python 3 Google Compute Engine backend (GPU)".

# Using Alphafold2 ColabFold - Input

- Input protein sequence, file name, relaxation, and template mode

▼ Input protein sequence(s), then hit **Runtime** -> **Run all**

```
[ ] #@title Input protein sequence(s), then hit `Runtime` -> `Run all`
from google.colab import files
import os
import re
import hashlib
import random

from sys import version_info
python_version = f"{version_info.major}.{version_info.minor}"

def add_hash(x,y):
    return x+"_"+hashlib.sha1(y.encode()).hexdigest()[:5]

query_sequence = 'MPRLSLLLPLLLLLLLPLLPPLSPSLGIRDVGGRRPKCGPCRPEGCPAPAPCP
#@markdown - Use `:` to specify inter-protein chainbreaks for **model1
jobname = 'test' #@param {type:"string"}
# number of models to use

print("jobname",jobname)
print("sequence",query_sequence)
print("length",len(query_sequence.replace(":", "")))
```


...

Choose Files No file chosen

Cancel upload

query\_sequence: "MPRLSLLLPLLLLLLLPLLPPLSPSLGIRDVGGRRPK" 

- Use **:** to specify inter-protein chainbreaks for **modeling complexes** (supports homo- and hetro-oligomers). For example **PI...SK:PI...SK** for a homodimer

jobname: "test" 

num\_relax: 0 

- specify how many of the top ranked structures to relax using amber

template\_mode: none 

- **none** = no template information is used. **pdb100** = detect templates in **pdb100** (see [notes](#)). **custom** - upload and search own templates (PDB or mmCIF format, see [notes](#))

# Using Alphafold2 ColabFold – Customizations

- Increasing recycles can lead to vastly increased structure quality

```
##@markdown ### Advanced settings
model_type = "auto" #@param ["auto", "alphafold2_ptm", "alphafold2_multimer_v1", "alphafold2_multimer_v2", "alphafold2_multimer_v3", "deepfold_v1", "alpha
##@markdown - if 'auto' selected, will use 'alphafold2_ptm' for monomer prediction and 'alphafold2_multimer_v3' for complex prediction.
##@markdown Any of the mode_types can be used (regardless if input is monomer or complex).
num_recycles = "3" #@param ["auto", "0", "1", "3", "6", "12", "24", "48"]
##@markdown - if 'auto' selected, will use 'num_recycles=20' if 'model_type=alphafold2_multimer_v3', else 'num_recycles=3'.
recycle_early_stop_tolerance = "auto" #@param ["auto", "0.0", "0.5", "1.0"]
##@markdown - if 'auto' selected, will use 'tol=0.5' if 'model_type=alphafold2_multimer_v3' else 'tol=0.0'.
relax_max_iterations = 200 #@param [0, 200, 2000] (type:"raw")
##@markdown - max amber relax iterations, '0' = unlimited (AlphaFold2 default, can take very long)
pairing_strategy = "greedy" #@param ["greedy", "complete"] (type:"string")
##@markdown - 'greedy' = pair any taxonomically matching subsets, 'complete' = all sequences have to match in one line.
calc_extra_ptm = False #@param (type:"boolean")
##@markdown - return pairwise chain iptm/actiptm

##@markdown ### Sample settings
##@markdown - enable dropouts and increase number of seeds to sample predictions from uncertainty of the model.
##@markdown - decrease 'max_msa' to increase uncertainty
max_msa = "auto" #@param ["auto", "512:1024", "256:512", "64:128", "32:64", "16:32"]
num_seeds = 1 #@param [1,2,4,8,16] (type:"raw")
use_dropout = False #@param (type:"boolean")

num_recycles = None if num_recycles == "auto" else int(num_recycles)
recycle_early_stop_tolerance = None if recycle_early_stop_tolerance == "auto" else float(recycle_early_stop_tolerance)
if max_msa == "auto": max_msa = None

##@markdown ### Save settings
save_all = False #@param (type:"boolean")
save_recycles = False #@param (type:"boolean")
save_to_google_drive = False #@param (type:"boolean")
##@markdown - If the save_to_google_drive option was selected, the result zip will be uploaded to your Google Drive
dpi = 200 #@param (type:"integer")
##@markdown - set dpi for image resolution

if save_to_google_drive:
    from pydrive2.drive import GoogleDrive
    from pydrive2.auth import GoogleAuth
    from google.colab import auth
    from oauth2client.client import GoogleCredentials
    auth.authenticate_user()
    gauth = GoogleAuth()
    gauth.credentials = GoogleCredentials.get_application_default()
    drive = GoogleDrive(gauth)
    print("You are logged into Google Drive and are good to go!")

##@markdown Don't forget to hit 'Runtime' -> 'Run all' after updating the form.
```

## Advanced settings

model\_type:

- if auto selected, will use alphafold2\_ptm for monomer prediction and alphafold2\_multimer\_v3 for complex prediction. Any of the mode\_types can be used (regardless if input is monomer or complex).

num\_recycles:

- if auto selected, will use num\_recycles=20 if model\_type=alphafold2\_multimer\_v3, else num\_recycles=3.

recycle\_early\_stop\_tolerance:

- if auto selected, will use tol=0.5 if model\_type=alphafold2\_multimer\_v3 else tol=0.0.

relax\_max\_iterations:

- max amber relax iterations, 0 = unlimited (AlphaFold2 default, can take very long)

pairing\_strategy:

- greedy = pair any taxonomically matching subsets, complete = all sequences have to match in one line.

calc\_extra\_ptm: ☐

- return pairwise chain iptm/actiptm

## Sample settings

- enable dropouts and increase number of seeds to sample predictions from uncertainty of the model.
- decrease max\_msa to increase uncertainty

max\_msa:

num\_seeds:

use\_dropout: ☐

## Save settings

save\_all: ☐

save\_recycles: ☐

save\_to\_google\_drive: ☐

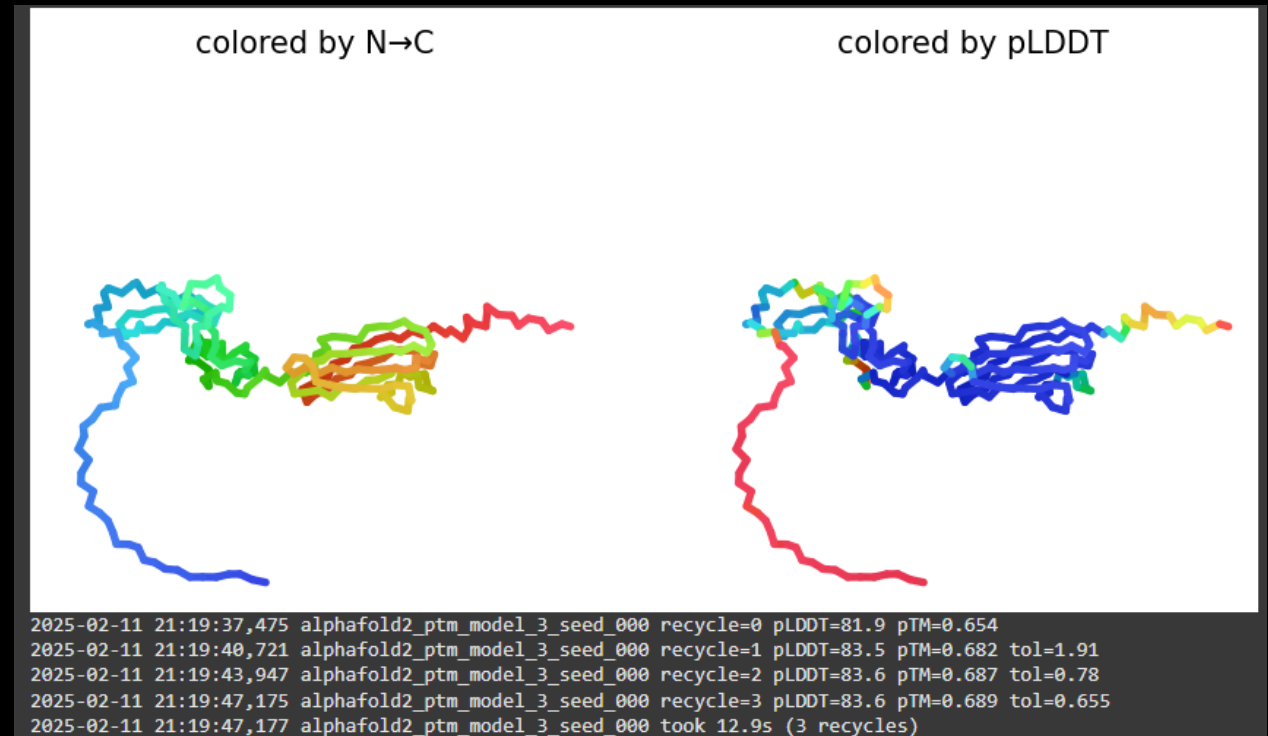
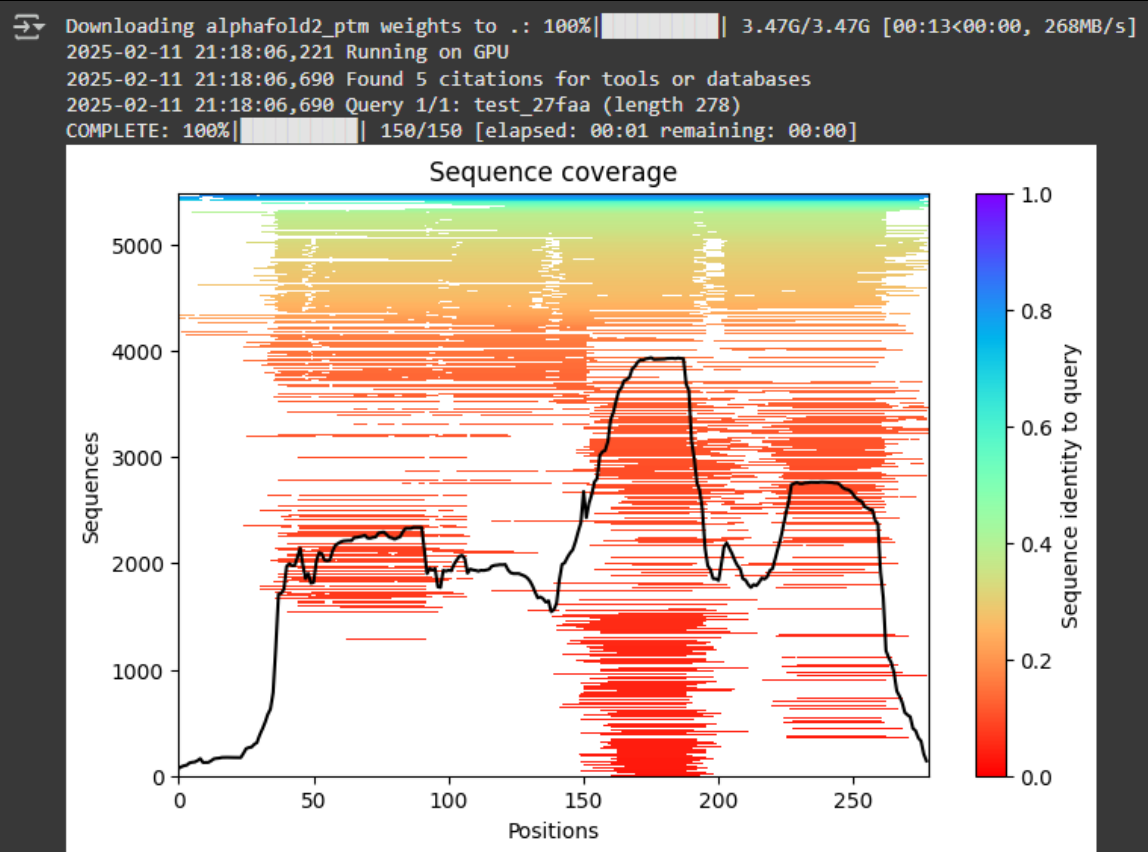
- if the save\_to\_google\_drive option was selected, the result zip will be uploaded to your Google Drive

dpi:

- set dpi for image resolution

# Using AlphaFold2 ColabFold – Output

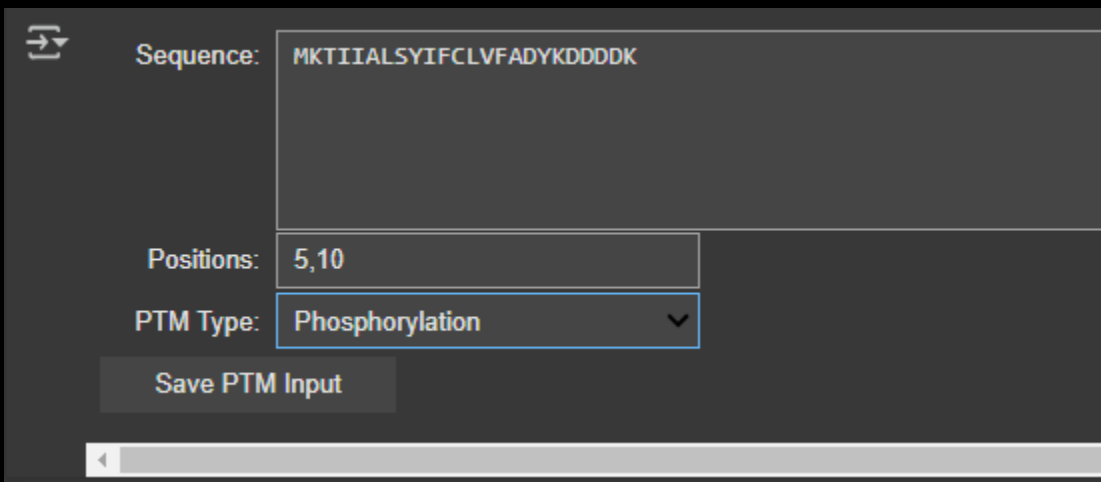
- MSA Sequence coverage chart and a series of structures with recycle data and confidence scores. **Structures output as PDB files.**



Confidence scores are reported as  
pLDDT and pTM

# Using Alphafold3 in a Colab Notebook

- Requires installation
- Requires some knowledge of python language
- Similar to using locally, but Colab handles most of the common causes for headache.



The screenshot shows the AlphaFold3 web interface. It has a dark theme. At the top left is a logo. The main form has three input fields: 'Sequence:' with the text 'MKTIIALSYIFCLVFADYKDDDDK', 'Positions:' with the text '5,10', and 'PTM Type:' with a dropdown menu showing 'Phosphorylation'. Below these fields is a button labeled 'Save PTM Input'. At the bottom of the form is a horizontal scrollbar.

```
# =====
# ⚙️ Step 2: Install AlphaFold 3
# =====

!git clone --quiet https://github.com/google-deepmind/alphafold3.git
!pip install -q --no-warn-conflicts -r alphafold3/requirements.txt

# Set model directory
os.environ['ALPHAFOLD3_MODEL_DIR'] = "/content/alphafold3/params"

print("✅ AlphaFold 3 installed successfully!")

# =====
# 📦 Step 3: Download Required Data
# =====
!alphafold3/fetch_databases.sh /content/alphafold3/data

print("✅ Databases downloaded successfully!")

# =====
# 🚀 Step 4: Run AlphaFold 3 Model
# =====

OUTPUT_DIR = "output"
DATA_DIR = "/content/alphafold3/data"

cmd = f"""
python alphafold3/run_alphafold.py \
  --input_json=input.json \
  --output_dir={OUTPUT_DIR} \
  --model_preset=multimer \
  --data_dir={DATA_DIR} \
  --use_gpu_relax=True
"""
```

# Using AlphaFold3 Server

- Very simple – Log in and input sequence
- Requires Google account, but is free to use
  - 20 uses per day
  - Jobs limited to 5,000 tokens (1 token ~ 1 residue)
- Lacks customization of recycles and MSA properties

The screenshot displays the AlphaFold3 Server web interface. At the top, the navigation bar includes 'AlphaFold Server' with a 'BETA' badge, and links for 'Server', 'About', and 'FAQ & Guides'. The 'Server' tab is active. In the top right corner, it shows 'Remaining jobs: 20'. Below this, a message states 'AlphaFold Server allows you to model a structure consisting of many biological molecules' with a 'Learn more' link. A list of features follows: 'Remaining jobs refresh each day', 'Jobs can be up to 5,000 tokens - see more details on token calculation, accepted formats, seed selection and other features in our FAQ', 'Use the entity bar to chemically modify proteins and nucleic acids', and 'Get in touch with the AlphaFold team if you have any questions'. A section titled 'Explore these examples of structures to see it in action – try them out without using your quota until you begin editing!' contains three buttons: 'Protein-RNA-Ion: PDB 8AW3', 'Protein-Glycan-Ion: PDB 7BBV', and 'Protein-DNA-Ion: PDB 7RCE'. Below this, there are 'Upload JSON' and 'Clear' buttons. The main input area features an 'Entity type' dropdown set to 'Protein', a 'Copies' field set to '1', and a large text input field with a placeholder '>Paste sequence or fasta' and the text 'Input'. A red error message 'This field is required' is visible below the input field. To the left of the input field is a grid icon, and to the right are three vertical dots and a chevron icon. At the bottom left is a '+ Add entity' button, and at the bottom right is a 'Save job' button. A 'Continue and preview job' button is centered at the very bottom.

# Using AlphaFold3 Server

- AF3 added the possibility to model non-protein entities
- Specify entity type in drop down menu.
- Specify number of copies

AlphaFold Server

BETA

Server

About

FAQ & Guides

Remaining jobs: 20

AlphaFold Server allows you to model a structure consisting of many biological molecules

Learn more

- Remaining jobs refresh each day
- Jobs can be up to 5,000 tokens - see more details on token calculation, accepted formats, seed selection and other features in our [FAQ](#)
- Use the entity bar to chemically modify proteins and nucleic acids
- Get in touch with the AlphaFold team if you have any questions

Explore these examples of structures to see it in action – try them out without using your quota until you begin editing!

Protein-Glycan-Ion: PDB 7BBV

Protein-DNA-Ion: PDB 7RCE

Upload JSON

Clear

> Paste sequence or fasta

Input

This field is required

Save job

Continue and preview job

Entity type

Protein

Protein

DNA

RNA

Ligand

Ion

Copies

1

>Paste sequence or fasta

Input

⋮

⌵

Copies

1

>Paste sequence or fasta

Input

⋮

⌵

Copies

1

>Paste sequence or fasta

Input

⋮

⌵

Copies

1

>Paste sequence or fasta

Input

⋮

⌵

+ Add entity

Save job

# Using AlphaFold3 Server

- AF3 also added PTM modeling.
- Select the PTM option to open a selection window.
- Very relevant for PPI assessment

The screenshot displays the AlphaFold3 Server interface. At the top, the navigation bar includes 'AlphaFold Server BETA', 'Server', 'About', and 'FAQ & Guides'. A status bar on the right indicates 'Remaining jobs: 20'. The main content area features a list of instructions: 'AlphaFold Server allows you to model a structure consisting of many biological molecules', 'Remaining jobs refresh each day', 'Jobs can be up to 5,000 tokens - see more details', 'Use the entity bar to chemically modify proteins', and 'Get in touch with the AlphaFold team if you have any questions'. Below this is a section titled 'Explore these examples of structures to see it in action'. A red circle highlights a three-dot menu icon in the 'Entity type' dropdown of the 'Protein' entity. This menu is expanded, showing options for 'Protein' (selected), 'DNA', 'RNA', 'Ligand', and 'Ion'. A red arrow points from this menu to a 'Post-Translational Modifications' window. This window contains a protein sequence with various residues highlighted in different colors (yellow, pink, orange, purple). Below the sequence, there are four rows for PTM selection, each with a colored box, a 'Type' dropdown, and a 'Select PTM' button. The first row is yellow (25 S), the second is pink (29 R), the third is orange (37 K), and the fourth is purple (113 T). The 'Type' dropdown for the first row is open, showing 'Phosphoserine' and 'Glycan chain' as options. At the bottom of the PTM window are 'Cancel' and 'Save' buttons.

AlphaFold Server BETA Server About FAQ & Guides

Remaining jobs: 20

AlphaFold Server allows you to model a structure consisting of many biological molecules [Learn more](#)

- Remaining jobs refresh each day
- Jobs can be up to 5,000 tokens - see more details
- Use the entity bar to chemically modify proteins
- Get in touch with the AlphaFold team if you have any questions

Explore these examples of structures to see it in action

Entity type: Protein (selected) | Copies: 1 | >Paste sequence or fasta Input

Entity type: Protein (selected) | Copies: 1 | >Paste sequence or fasta Input

Entity type: DNA | Copies: 1 | >Paste sequence or fasta Input

Entity type: RNA | Copies: 1 | >Paste sequence or fasta Input

Entity type: Ligand | Copies: 1 | >Paste sequence or fasta Input

Entity type: Ion | Copies: 1 | >Paste sequence or fasta Input

+ Add entity Save job

Post-Translational Modifications

Once you add PTMs and save it, you can't edit the sequence [Learn more](#)

Protein sequence: MPRLSLLPL LLLLLLPLP PLSPSLGIID VGGRRPCGP CRPEGCPAPA PCPAPGISAL DEGCCARCL  
GAEGASCGGR AGGRCGPGLV CASQAAGAAP EGTGLCVCAQ RGVCVGSDDR SYPSVICALRL RARHTPRAHP  
GHLHKARDGP CEFAPVVVVP PRSVHNVTA QVGLSCEVRA VTPVITWRK VTKSPEGTQA LEELPGDHVN  
IAVQVRGGPS DHEATAWILI NPLRKEDEGV YQCHAANMVG EAESHSTVTV LDLSKYRSFH FPAPDDR

PTM Selection:

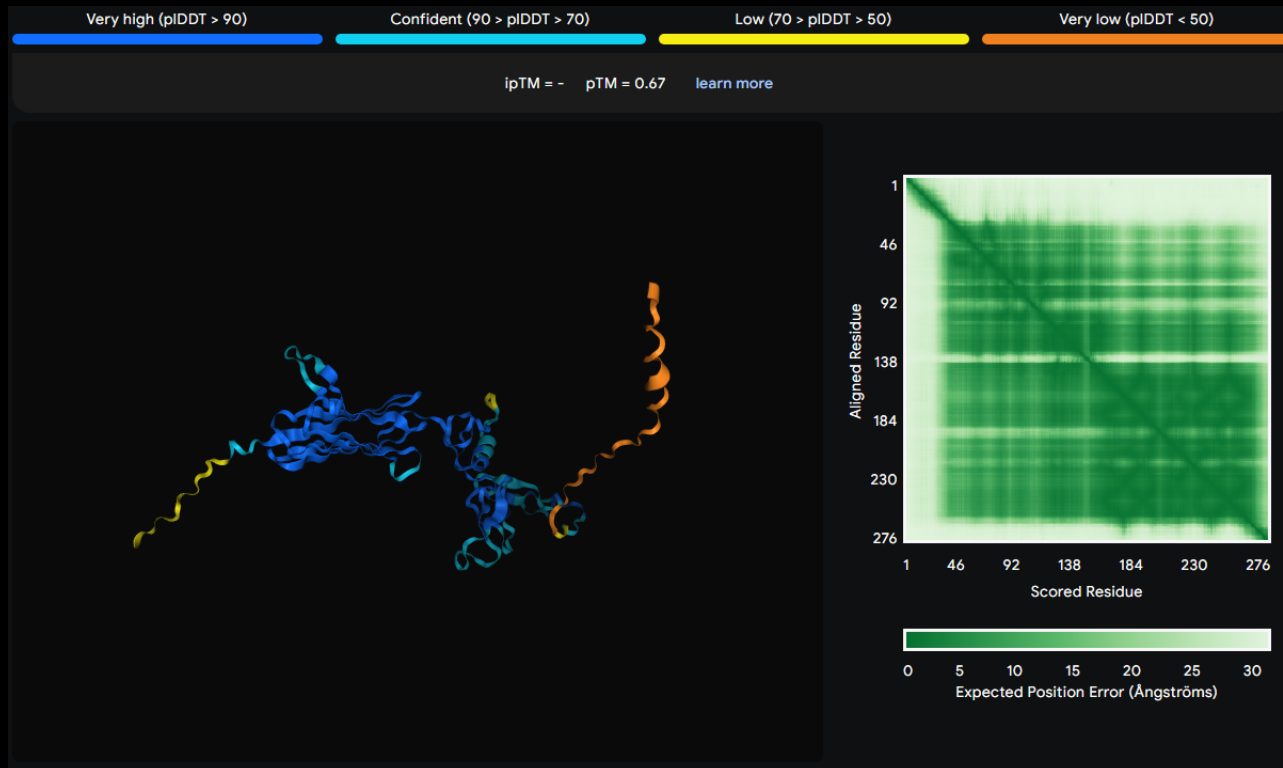
- 25 S Type: Select PTM
- 29 R Type: Select PTM
- 37 K Type: Select PTM
- 113 T Type: Select PTM

Cancel Save

# Using AlphaFold3 Server - Output

- Similar to ColabFold
  - (Confidence scores and PDB files)
- Need special software to open PDB

```
"fraction_disordered": 0.16,  
"has_clash": 0.0,  
"iptm": 0.92,  
"num_recycles": 10.0,  
"ptm": 0.78,  
"ranking_score": 0.97
```



cat-cadherin.pdb											
File Edit View											
ATOM	1	N	ILE	A	1	-65.952	21.848	82.645	1.00	46.67	A N
ATOM	2	CA	ILE	A	1	-66.828	21.079	81.738	1.00	53.02	A C
ATOM	3	C	ILE	A	1	-66.435	21.303	80.280	1.00	55.51	A C
ATOM	4	O	ILE	A	1	-66.245	20.328	79.562	1.00	51.42	A O
ATOM	5	CB	ILE	A	1	-68.322	21.376	81.992	1.00	50.67	A C
ATOM	6	CG1	ILE	A	1	-68.706	20.981	83.438	1.00	46.78	A C
ATOM	7	CG2	ILE	A	1	-69.202	20.610	80.975	1.00	46.21	A C
ATOM	8	CD1	ILE	A	1	-70.138	21.375	83.842	1.00	41.25	A C
ATOM	9	N	ALA	A	2	-66.269	22.560	79.866	1.00	47.12	A N
ATOM	10	CA	ALA	A	2	-65.848	22.876	78.495	1.00	53.42	A C
ATOM	11	C	ALA	A	2	-64.479	22.278	78.143	1.00	54.07	A C
ATOM	12	O	ALA	A	2	-64.316	21.708	77.067	1.00	53.80	A O
ATOM	13	CB	ALA	A	2	-65.849	24.400	78.319	1.00	52.70	A C
ATOM	14	N	ILE	A	3	-63.526	22.330	79.084	1.00	53.57	A N
ATOM	15	CA	ILE	A	3	-62.201	21.716	78.898	1.00	57.37	A C
ATOM	16	C	ILE	A	3	-62.327	20.193	78.800	1.00	58.40	A C
ATOM	17	O	ILE	A	3	-61.719	19.590	77.919	1.00	58.22	A O
ATOM	18	CB	ILE	A	3	-61.233	22.141	80.018	1.00	55.84	A C
ATOM	19	CG1	ILE	A	3	-61.009	23.672	79.983	1.00	52.70	A C
ATOM	20	CG2	ILE	A	3	-59.889	21.405	79.892	1.00	51.46	A C
ATOM	21	CD1	ILE	A	3	-60.210	24.222	81.175	1.00	47.43	A C
ATOM	22	N	LEU	A	4	-63.161	19.570	79.652	1.00	52.54	A N
ATOM	23	CA	LEU	A	4	-63.443	18.133	79.579	1.00	55.85	A C
ATOM	24	C	LEU	A	4	-64.131	17.756	78.266	1.00	56.08	A C
ATOM	25	O	LEU	A	4	-63.759	16.756	77.664	1.00	58.52	A O
ATOM	26	CB	LEU	A	4	-64.290	17.704	80.788	1.00	54.75	A C
ATOM	27	CG	LEU	A	4	-63.441	17.080	81.906	1.00	52.17	A C
ATOM	28	CD1	LEU	A	4	-64.108	17.263	83.261	1.00	46.73	A C
ATOM	29	CD2	LEU	A	4	-63.233	15.592	81.663	1.00	51.22	A C
ATOM	30	N	LEU	A	5	-65.097	18.582	77.803	1.00	55.32	A N
ATOM	31	CA	LEU	A	5	-65.719	18.397	76.488	1.00	58.56	A C
ATOM	32	C	LEU	A	5	-64.704	18.546	75.357	1.00	59.04	A C
ATOM	33	O	LEU	A	5	-64.708	17.720	74.452	1.00	59.95	A O
ATOM	34	CB	LEU	A	5	-66.887	19.381	76.307	1.00	57.00	A C
ATOM	35	CG	LEU	A	5	-68.231	18.783	76.751	1.00	52.99	A C
ATOM	36	CD1	LEU	A	5	-69.238	19.887	77.060	1.00	48.19	A C
ATOM	37	CD2	LEU	A	5	-68.813	17.888	75.666	1.00	51.41	A C
ATOM	38	N	CYS	A	6	-63.824	19.543	75.442	1.00	61.27	A N
ATOM	39	CA	CYS	A	6	-62.741	19.674	74.465	1.00	63.73	A C
ATOM	40	C	CYS	A	6	-61.791	18.478	74.505	1.00	63.19	A C
ATOM	41	O	CYS	A	6	-61.422	17.979	73.449	1.00	64.83	A O
ATOM	42	CB	CYS	A	6	-61.968	20.976	74.685	1.00	62.96	A C
ATOM	43	SG	CYS	A	6	-62.890	22.371	73.985	1.00	55.84	A S
ATOM	44	N	ILE	A	7	-61.427	17.988	75.703	1.00	61.19	A N
ATOM	45	CA	ILE	A	7	-60.582	16.791	75.838	1.00	62.94	A C
ATOM	46	C	ILE	A	7	-61.316	15.553	75.318	1.00	62.47	A C
ATOM	47	O	ILE	A	7	-60.706	14.746	74.622	1.00	64.37	A O

Ln 25, Col 81 | 1,402,466 characters | 100% | Unix (LF) | UTF-8

# Output – Confidence Scores

**pLDDT** – predicted local distance difference test

- Local, residue-level confidence score
- Can provide indications of disordered or unstructured regions

Ranges from 0 to 100

- >90: Very high confidence

**pTM** – predicted template modeling

- Global, structure-level confidence score

**piTM** – predicted interface template modeling

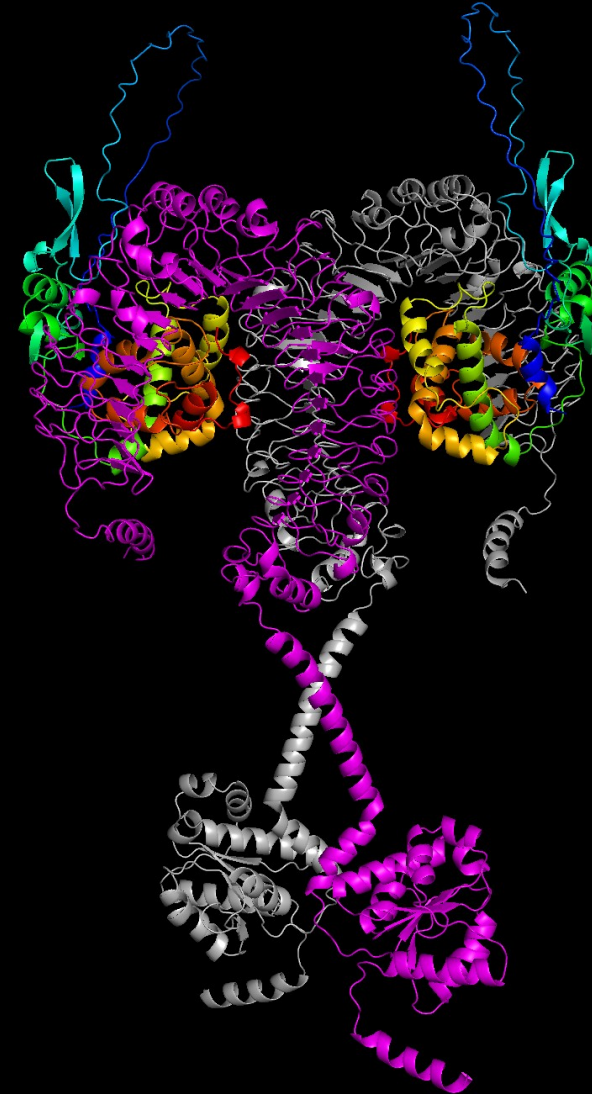
- Evaluation of how well chains interact at protein interfaces

pTM and piTM range from 0 to 1

- >0.6: Likely a real structure

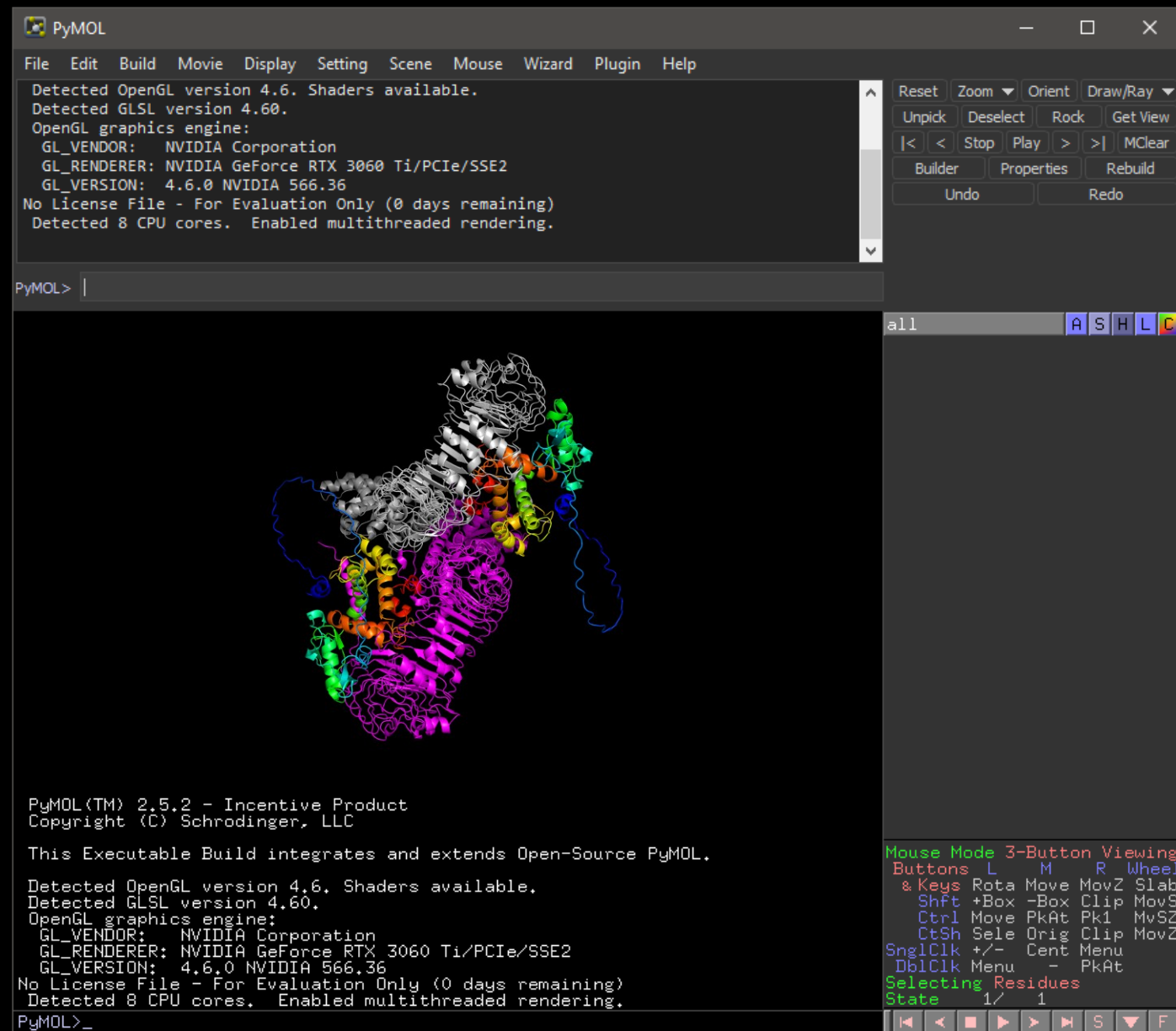
# Output – PDB Files: Visualizing your structures

- Need molecular visualization software to open files
- PyMOL and ChimeraX are the most common choices
- Both are free to use and work with Windows/MacOS
- Best to have both on hand



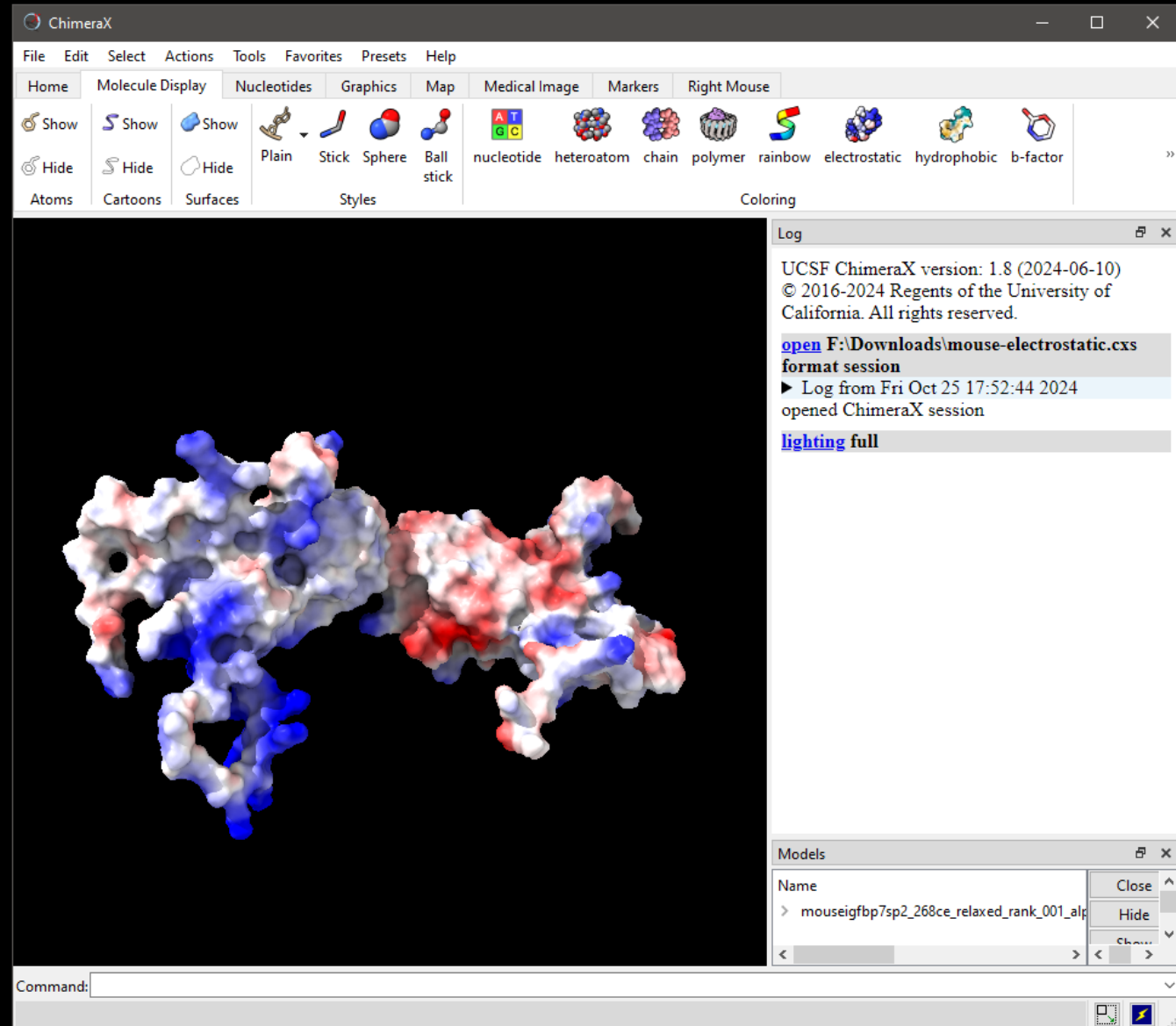
# PyMOL

- Open-source, Python-based molecular visualization tool
- Utilizes a Python-based command line for most operations
- Most functions/commands are covered in the PyMOL wiki
- Not as flashy as ChimeraX, but more granular



# UCSF-ChimeraX

- Open-access molecular visualization software developed by UCSF
- Utilizes a GUI for most features
  - Proprietary command list and .py scripts are also used in some cases
- ChimeraX wiki is comprehensive and covers commands in detail
- Excels at figure design



# Next Steps – Analyze your structure

- Further analyses can be conducted on your structure
- Servers and software packages are typically purpose-built and rather specific
- Some analyses are user-friendly GUIs and others are command-line based
- **Google is your friend!**



DockQ

PS-PRED

CABSflex

mCSM-PPI2

DDMut-PPI

AutoDock

flare

CABS-dock server for flexible protein-peptide docking

# Summary

- Protein structures can reveal a lot about their functions
- AlphaFold is a highly-accessible tool for predicting protein and protein complex structures
- PyMOL and ChimeraX can be used to freely visualize and analyze structures
- Assay-specific downstream software is readily available

**February 26<sup>th</sup>, 2025**

- Downstream software examples
- PPI prediction and manipulation
- DockQ analysis