“RCSB-PDB– Bio Python Pipeline”

Bio Python (BTY-162)

Report submitted to the

School of Bioengineering and Biosciences

For the partial fulfilment for the award of the

Degree of

BACHELOR OF TECHNOLOGY

In

BIOTECHNOLOGY Session 2024-2025

By

Satvik Kunigiri

12319419

Under the Guidance of

Dr. Piyush Kumar Yadav

Department of Biotechnology

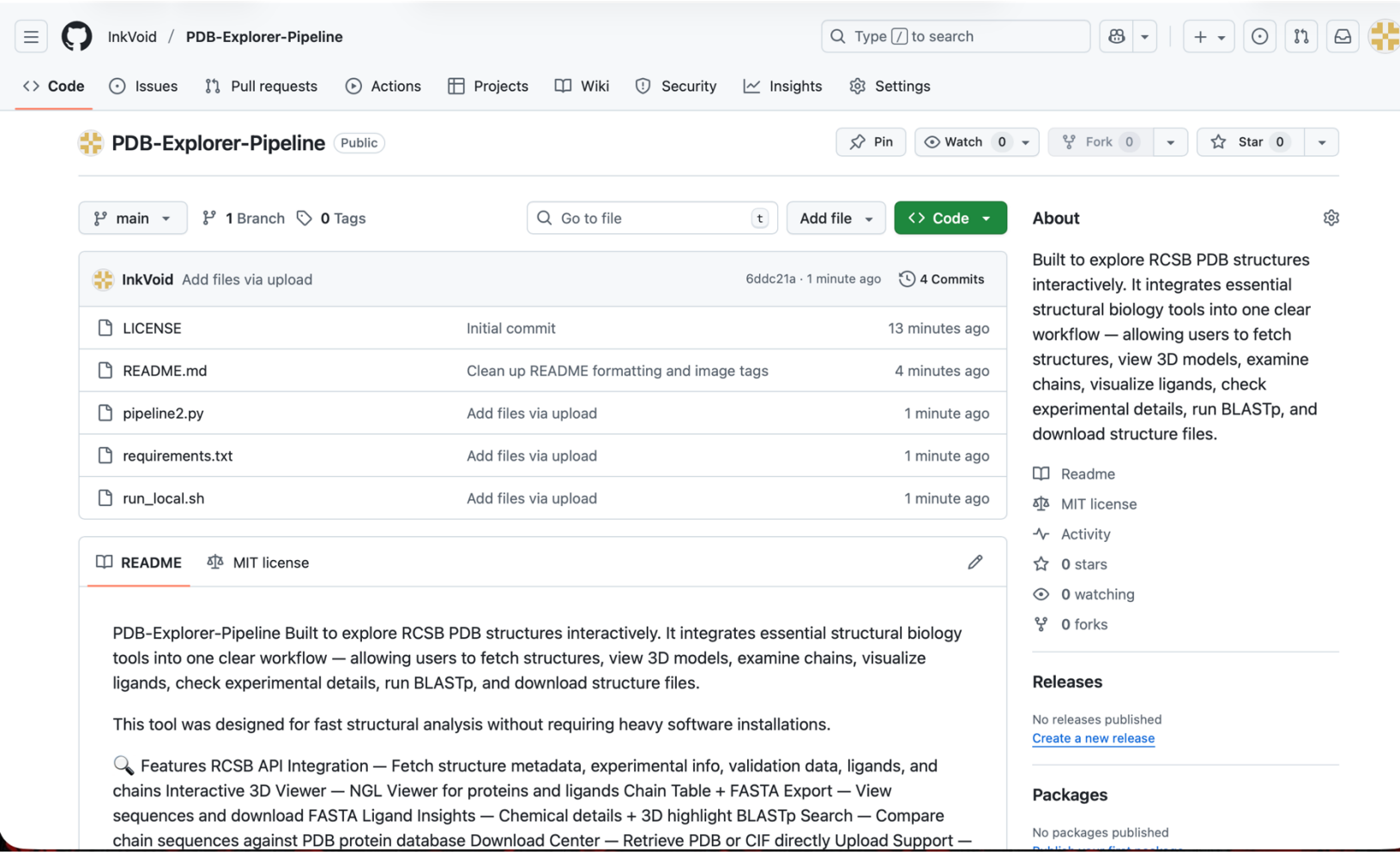
SCHOOL OF BIOENGINEERING AND BIOSCIENCES

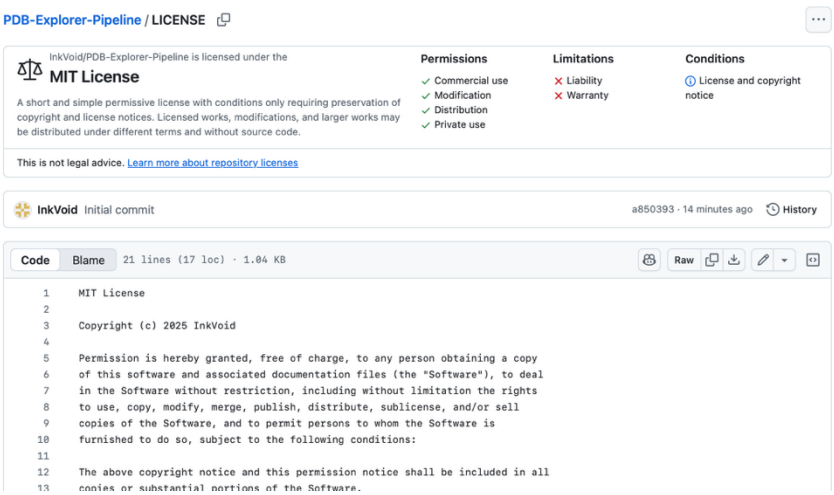
LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

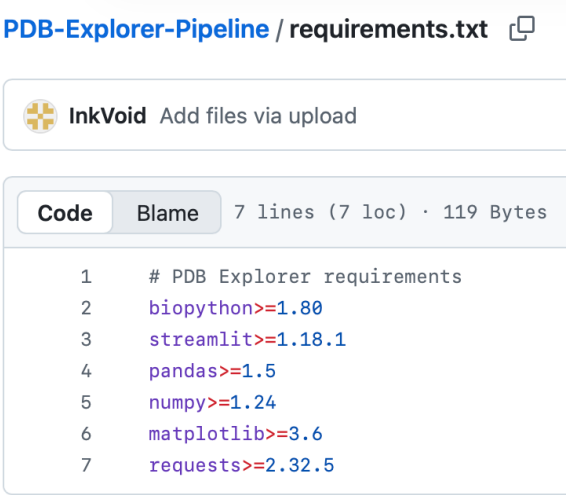
GITHUB LINK: *https://github.com/InkVoid/PDB-Explorer-Pipeline/tree/main*

Mention: ***badabheem (Piyush Sir)***

Upload Proof of REPO:





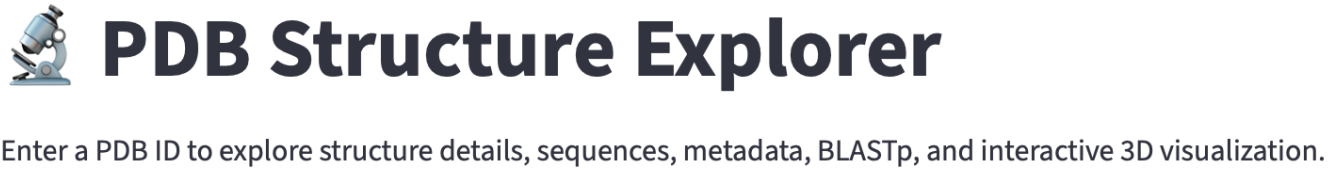


Introduction:

Bioinformatics has become an essential discipline in modern biological research, bridging computational techniques with molecular biology to analyze and interpret biological data. The rapid growth of nucleotide and protein sequence databases has made it increasingly important to design tools that simplify preliminary sequence analysis while remaining adaptable to advanced tasks.

The RCSB PDB pipeline provides a structured and automated framework for accessing, organizing, and preparing biological macromolecular structures for computational workflows. Modern structural biology relies heavily on large datasets of high-quality protein and nucleic acid structures, yet manual retrieval and curation from the Protein Data Bank can be time-consuming, inconsistent, and prone to human error. A dedicated pipeline resolves these challenges by integrating automated querying, metadata filtering, file standardization, and basic quality checks into a single streamlined process.

By doing so, it ensures that only relevant, reproducible, and analysis-ready structures enter downstream stages such as molecular simulations, structural alignment, functional annotation, or machine-learning-based feature extraction. This level of organization is especially important when dealing with hundreds or thousands of PDB entries, where uniformity directly affects the reliability of conclusions. Ultimately, the RCSB PDB pipeline acts as the foundation of a robust computational research ecosystem, enabling faster exploration, clearer documentation, and more rigorous scientific outcomes.



THIS REPORT WILL BE TESTING THE PROTEIN:

4O4S
Crystal structure of phycobiliprotein lyase CpcT complexed with phycocyanobilin (PCB)

CODE Explanation and Working:

This imports all the necessary **Biopython modules.** **Streamlit** is used for the web interface. The **PDB API** allows querying the RSCB databases with just an accession number.


```python
import streamlit as st
import requests
import pandas as pd
import base64
from Bio import Entrez
from Bio.Blast import NCBIWWW, NCBIXML
from html import escape
from io import StringIO
```

```python
# ---------- Helper: RCSB endpoints ----------
def rcsb_get_json(url):  6 usages
    try:
        r = requests.get(url, timeout=20)
        r.raise_for_status()
        return r.json()
    except Exception:
        return None

def fetch_entry(pdb_id):  1 usage
    return rcsb_get_json(f"https://data.rcsb.org/rest/v1/core/entry/{pdb_id}")

def fetch_struct_summary(pdb_id):  1 usage
    return rcsb_get_json(f"https://data.rcsb.org/rest/v1/core/structure_summary/{pdb_id}")

def fetch_polymer_entity(pdb_id, entity_id):  1 usage
    return rcsb_get_json(f"https://data.rcsb.org/rest/v1/core/polymer_entity/{pdb_id}/{entity_id}")

def fetch_chem_comp(chem_id):  1 usage
    return rcsb_get_json(f"https://data.rcsb.org/rest/v1/core/chem_comp/{chem_id}")

def fetch_validation(pdb_id):  1 usage
    return rcsb_get_json(f"https://data.rcsb.org/rest/v1/core/validation_summary/{pdb_id}")

def fetch_experiment(pdb_id):  1 usage
    return rcsb_get_json(f"https://data.rcsb.org/rest/v1/core/experiment/{pdb_id}")
```

Usage of Helper Endpoints from RCSB

INPUT SECTION:

PDB ID (e.g. 4O4S)

Or upload PDB/mmCIF

Drag and drop file here

Limit 200MB per file • PDB, CIF, …

Browse files

SUMMARY TABLE:

Structure Overview

	Field	Value
0	PDB ID	4O4S
1	Entry ID	4O4S
2	Title	Crystal structure of phycobiliprotein lyase CpcT complexed with phycocyanobilin (PCB)
3	Experimental Method	X-RAY DIFFRACTION
4	Release Date	2014-08-13T00:00:00+0000
5	Deposition Date	2013-12-19T00:00:00+0000

Experimental Data

Method: X-RAY DIFFRACTION
Resolution: 2.50 Å
R-Value Free: 0.248 (Depositor), 0.240 (DCC)
R-Value Work: 0.180 (Depositor), 0.180 (DCC)
R-Value Observed: 0.184 (Depositor)

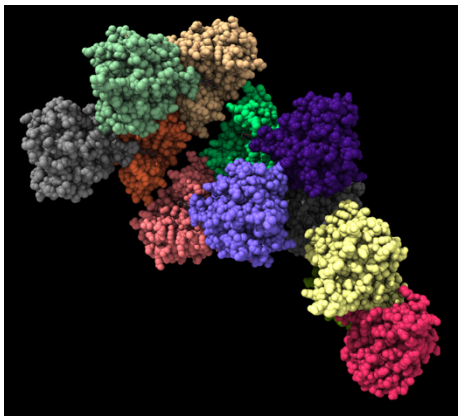
Space Group: P 1

Unit Cell:

Length (Å)	Angle (°)
a = 69.746	α = 90.21
b = 69.601	β = 90.28
c = 162.595	γ = 60.12

Software Package:

Software Name	Purpose
MAR345dtb	data collection
PHASER	phasing
PHENIX	refinement
HKL-2000	data reduction
HKL-2000	data scaling



Core Parts:

```python
# ---------- Page: BLAST ----------
elif menu == "BLAST":
    st.header("BLASTp (protein vs PDB)")
    st.markdown("Select a chain sequence (Chains page) or paste a protein sequence and run BLASTp against the PDB database.")
    # sequence input area
    seq_area = st.text_area("Protein sequence (FASTA or raw). If left empty, selected chain sequence will be used.", value="")
    # allow user to pick selected chain from session
    if st.session_state.poly_seqs:
        default_choice = st.session_state.selected_entity or list(st.session_state.poly_seqs.keys())[0]
        chosen_chain = st.selectbox("Or choose chain sequence", options=["(use text area)"] + list(st.session_state.poly_seqs.key
    else:
        chosen_chain = "(use text area)"
    blast_btn = st.button("Run BLASTp (top 5)")
```

BLASTp to run against only the protein structures and provides a table of results

```python
def ngl_viewer_html_from_url(url, show_hetero=True, highlight_resname=None):  3 usages
    # url is a direct fetchable PDB file url or data URL
    # highlight_resname: e.g. "HEM" to highlight ligand
    # escape braces for f-string JS blocks
    sel_high = f'var highlightSel = "resname {highlight_resname}";' if highlight_resname else "var highlight
    hetero_repr = 'o.addRepresentation("ball+stick", {sele: "hetero", color: "element"});' if show_hetero el
    html = f"""
```

The NGL viewer MODULE, that is responsible for the 3D visualization.

```python
# ---------- Page: Overview ----------
if menu == "Overview":
    st.header("Overview")
    st.markdown("Main structure-level fields from RCSB (summary).")
    if st.session_state.pdb_source == "rcsb" and st.session_state.entry_json:
        entry = st.session_state.entry_json
        # try to gather straightforward summary fields
        title = entry.get("struct", {}).get("title", "")
        methods = ", ".join([e.get("method", "") for e in entry.get("exptl", [])]) if entry.get("exptl") else ""
        release = entry.get("rcsb_accession_info", {}).get("initial_release_date", "")
        deposition = entry.get("rcsb_accession_info", {}).get("deposit_date", "")
        rev = entry.get("rcsb_accession_info", {}).get("revision_date", "")
        pdb_info = [
            ["PDB ID", pdb_input],
            ["Title", title],
            ["Experimental Method(s)", methods],
            ["Release date", release],
            ["Deposition date", deposition],
            ["Revision date", rev],
            ["Source", "RCSB Data API"]
        ]
```

🧩 Macromolecules (Chains)

	Entity ID	Length	Type	Strand ID
0	1	207	polypeptide(L)	A,B,I,J,C,D,E,F,G,H,K,L

View Sequence (select entity)

1

```
MTHSTDIATLARWMAADFSNQAQAFENPPFYAHIRVCMRPLPWEVLSGVGFFVEQAYDYMLNDPYRLRVLKLMIVGDRIHIENYTVKQEENFYGASRDLNRLQTLTSESLEKLPGCNMIVEWTGNSFKGTVEPGKGCIVVRK
```

Download selected chain as FASTA

🧪 Experimental Details

	Metric	Value
0	Methods	X-RAY DIFFRACTION
1	Resolution (Å)	2.5
2	Space Group	P 1

SIDEBAR & SETTINGS:

Controls

Page

- (•) Overview
- ( ) Visualization
- ( ) Chains
- ( ) Ligands
- ( ) Experimental
- ( ) Validation
- ( ) BLAST
- ( ) Downloads

NCBI email (required for BLAST)

Enter a valid email for NCBI qblast. No email → BLAST blocked.

EMAIL is mandatory to access the ENTREZ database and usage of the BLASTp Tool.

.

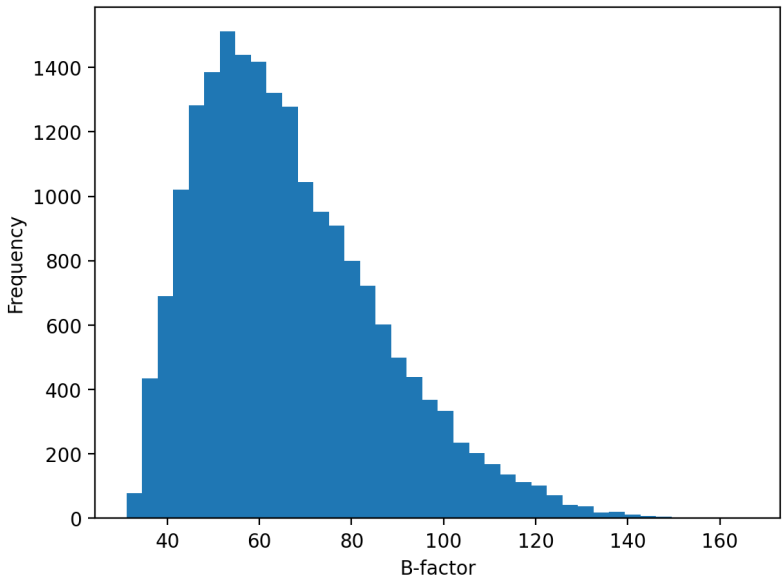
Coordinate file analysis

Geometric and biophysical analysis of the loaded coordinate file (PDB/mmCIF). Requires a parsed structure (auto-parsed after fetch/upload if possible).

Atom table (first 500 rows)

	model	chain	residue	res_id	atom	x	y	z	b_factor	is_ligand
0	0	A	THR	2	N	-0.3580000102519989	7.581999778747559	-55.694000244140625	122.6	False
1	0	A	THR	2	CA	-0.12800000607967377	8.788000106811523	-54.909000396728516	120.55	False
2	0	A	THR	2	C	-1.4500000476837158	9.37399959564209	-54.4119987487793	122.55	False
3	0	A	THR	2	O	-2.433000087738037	9.449999809265137	-55.1619987487793	75.41	False
4	0	A	THR	2	CB	0.6480000019073486	9.857999801635742	-55.71200180053711	114.14	False
5	0	A	THR	2	OG1	-0.1889999955892563	10.376999855041504	-56.744998931884766	117.66	False
6	0	A	THR	2	CG2	1.878000020980835	9.253999710083008	-56.347999572753906	110.08	False
7	0	A	HIS	3	N	-1.4620000123977661	9.779999732971191	-53.14400100708008	119.09	False
8	0	A	HIS	3	CA	-2.631999969482422	10.404999732971191	-52.53799819946289	117.01	False
9	0	A	HIS	3	C	-2.2079999446868896	11.418999671936035	-51.474998474121094	111.02	False

Download full atom table (CSV)



Cα pairwise distances (n=2370)

Cα count 2370 exceeds 500. Skipping full distance heatmap to avoid browser lag.

PROVISION of Direct Download:

Downloads

Download legacy PDB or mmCIF files directly from RCSB (if using a fetched PDB ID). You can also download selected chain FASTA.

- Download PDB (.pdb)
- Download mmCIF (.cif)

BLASTn:

```
135    def run_blastn_top(seq_text, top=BLAST_MAX_HITS):  1 usage
136        res = NCBIWWW.qblast("blastn","nt", seq_text)
137        hits = []
138        for r in NCBIXML.parse(res):
139            for a in r.alignments[:top]:
140                h = a.hsps[0] if a.hsps else None
141                hits.append({"title":a.title, "len":a.length, "score": h.score if h else None, "e": h.expect if h else None})
142            break
143        res.close()
144        return hits
```

The pipeline runs BLASTp against the nucleotide database and limits the output to the top 5 hits. It works through the NCBIWWW, where it allows python to conduct BLAST remotely.

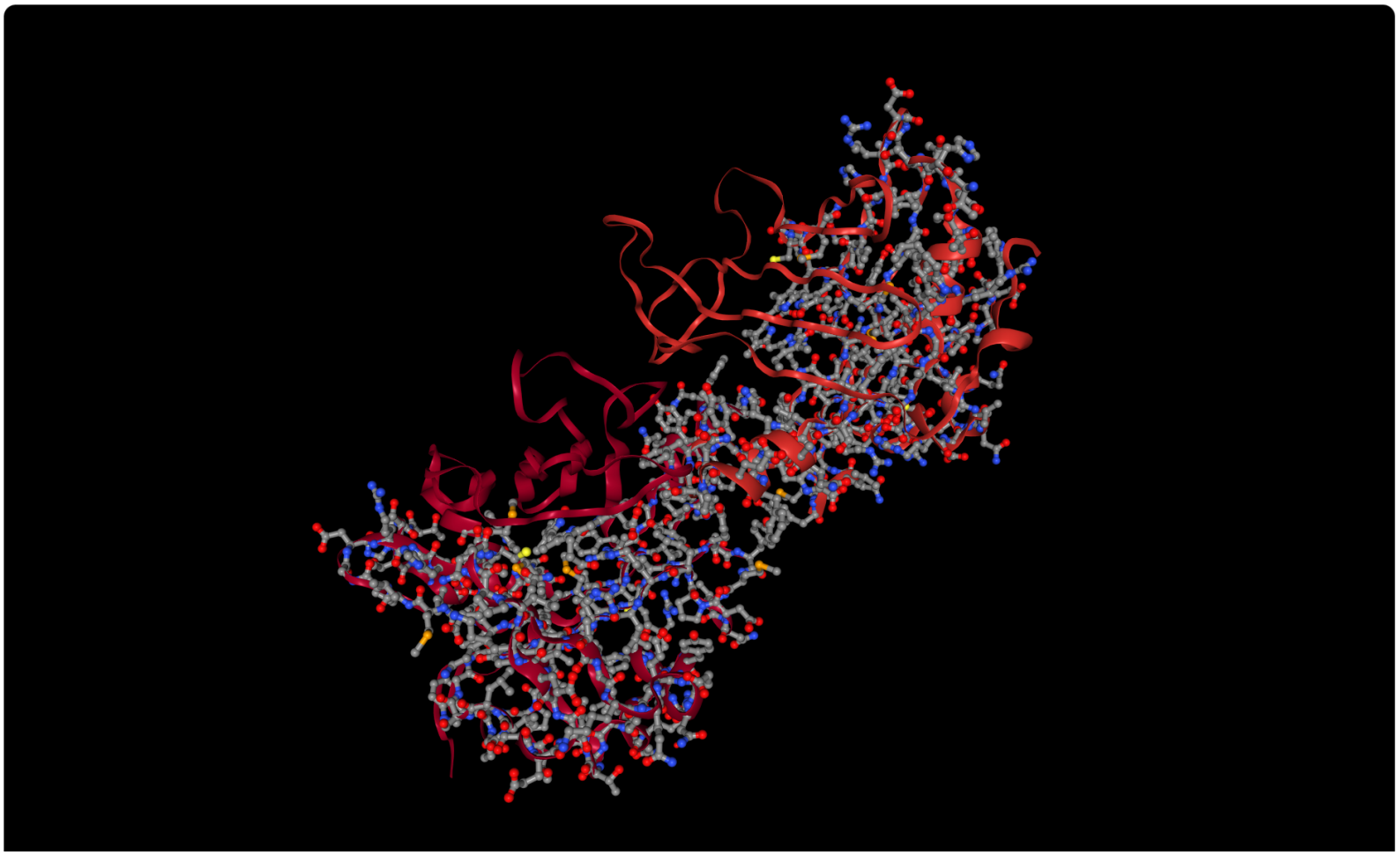
PDB:

Streamlit Interface: Automatically updates outputs (tables, plots, metrics, and sequence displays) when a user interacts with the UI. This keeps the application lightweight and user-friendly.

RESULTS:

Visualization

Interactive 3D view. You can highlight ligands (if any).



Conclusion:

Developing an RCSB PDB pipeline not only improves workflow efficiency but also elevates the scientific rigor of structural research. By formalizing how structures are selected, downloaded, cleaned, and prepared, the pipeline minimizes inconsistencies and ensures that each dataset is fully traceable and reproducible. This leads to more reliable simulations, better quality control in large-scale analyses, and easier collaboration across research teams.

The pipeline also serves as a scalable tool equally suited for targeted studies on a small protein family or extensive projects involving thousands of structures. Its automation reduces the cognitive load on researchers, allowing them to focus on interpretation, hypothesis generation, and innovation rather than repetitive data handling. In this way, the RCSB PDB pipeline becomes more than a technical utility; it becomes an essential part of a mature, transparent, and modern computational biology workflow.

THANK YOU