Team Red: Srashti Bajpai, Qiyi Yu, Tony Tsai, Xianda Ma, Manpreet Kaur

Protein Analysis and Docking Pipeline

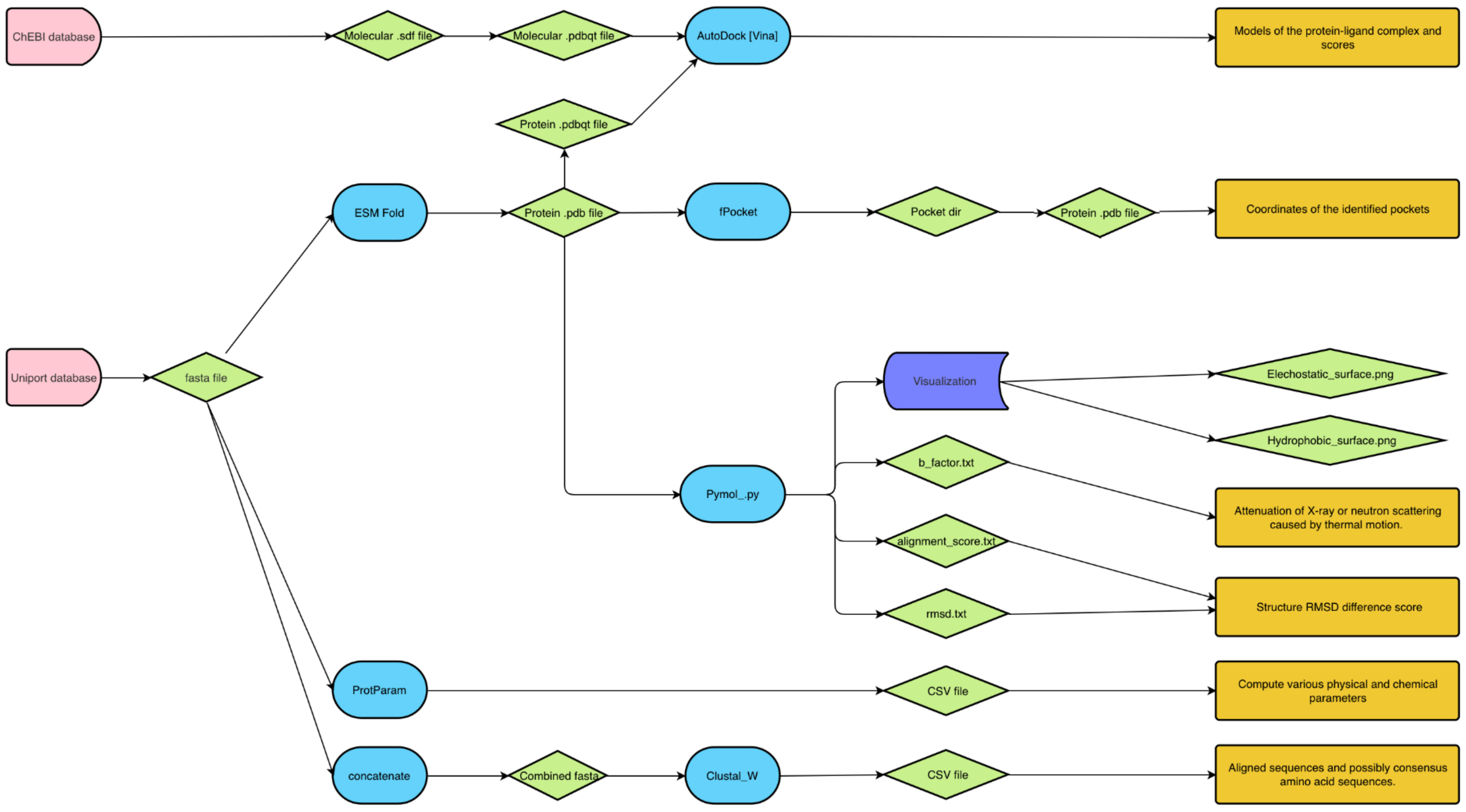
Table of Contents:

Introduction

About this Pipeline

This pipeline is an integrated suite of bioinformatics tools specifically designed to enhance and simplify the study of protein structures and functions. This pipeline facilitates comprehensive analysis of protein sequences, from prediction and parametrization to structural alignments, visualization of binding pockets, preparing files for docking, and providing blind docking, between a protein and ligand of interest, supporting a wide range of biochemical and molecular biology research objectives. Utilizing ESMfold, the pipeline predicts the three-dimensional structure of proteins from their amino acid sequences. Through Protparam, the pipeline calculates various properties of the protein sequences such as molecular weight, isoelectric point, aromaticity, and instability index. Clustal-W is employed to align multiple protein sequences, enabling the identification of conserved regions and structural motifs. The pipeline also includes tools like fPockets to identify and visualize potential active sites or ligand-binding regions in the protein structures. It also includes converting .pdb files and .sdf files, into suitable format for running small-molecular blind docking using AutoDock Vina.

Workflow



Packages

1. ESMfold:

ESMfold (Evolutionary Scale Modeling for Protein Folding) is a software package that predicts the three-dimensional structures of proteins by utilizing evolutionary information from protein sequences.
Input: A protein sequence that can be obtained from the NCBI or UniProt database in .fasta format.
Output: A .pdb file containing the 3D structure information of the protein.
For installation, please refer:
https://github.com/facebookresearch/esm

2. ProtParam:

This tool is used to analyze and character protein sequences, which can provide insights into their structure, function, and potential interactions with other molecules.
Input: Protein sequence
Output: Protein ID, Molecular Weight, Isoelectric Point, Aromaticity,Instability Index, and Amino Acid.
For more details and installation, please refer:
https://biopython.org/docs/1.76/api/Bio.SeqUtils.ProtParam.html

3. Clustal-Omega:

Clustal Omega is a multiple sequence alignment program that uses seeded guide trees and HMM profile-profile techniques to generate alignments between three or more sequences.
Input: Protein sequence(fasta)
Output: multiple sequence alignment of the input sequences(fasta)
For installation, please refer:
https://github.com/facebookresearch/esm

4. fPockets:

This tool is used to analyze protein structures and identify potential binding sites or pockets within those structures. It is able to gain insights into the functional properties of proteins and use this information for drug discovery, protein engineering, and other molecular biology applications.
Input: PDB file
Output: Information about the detected pockets.
For installation, please refer:
https://github.com/facebookresearch/esm

5. AutoDock Vina:

AutoDock Vina is one of the most widely used open-source docking engines. It is a turnkey computational docking program that is based on a simple scoring function and rapid gradient-optimization conformational search.
Input: .pdbqt files for the receptor and ligand.
Output: .pdbqt files. The estimated binding energy of the ligand to the receptor, usually expressed in kcal/mol. This value predicts how strongly the ligand binds to the protein. Multiple conformations of the ligand as it binds to the receptor.
For installation, please refer:
https://github.com/facebookresearch/esm

Other required installations:

- ➔ pip install termcolor
- ➔ pip install biopython
- ➔ conda install -c conda-forge rdkit (https://www.rdkit.org/docs/Install.html)
- ➔ Open Babel (obabel) (https://openbabel.org/docs/index.html)

Running Pipeline

Required Input Files:

1. ***.fasta files:*** Used by ESMfold, ProtParam, and Clustal-Omega
2. ***.pdb files:*** These will be generated by ESMfold, and used as input by fPockets, and for prepping of files for running AutoDock Vina.
3. ***.sdf files:*** Will be used for prepping of files for running of AutoDock Vina

These files can be found on platforms like:
Protein sequence and RNA sequence(.fasta)
Uniport database: https://www.uniprot.org/
NCBI database: https://www.ncbi.nlm.nih.gov/protein/

Molecular file(.sdf)
ChemEBI database: https://www.ebi.ac.uk/chebi/init.do
PubChem database: https://pubchem.ncbi.nlm.nih.gov/

Usage:

All the scripts are provided in the main master folder ‘pipeline_scripts’, it contains esm_fold.py [for 3D structure prediction], pymol_.py [for structural analysis], main.py [runs ProtParam, Clustal-Omega, fPockets, and prepps the files for docking] sub-files for main.py are present in folder called py_files_for_main, and Docking_.py [runs AutoDock Vina, do not remove config.txt, as it provides reference for AutoDock].

The run_esmFold.py is supposed to be run on interact –gpu, and will be run by the below command with torch_home, and the .fasta file as arguments. Make sure that the script is running in the esmfold environment, otherwise the .py file might give an error.
conda activate esmfold

```
python esm_fold.py "path/to/torch_home" "/path/to/.fasta_file"
```

To find structural properties and visualise ESMfold, open the script pymol_.py and manually add the path to your .pdb file, as this script is hard coded, initiate pymol in the terminal and run the following command,

```
run /path/to/pymol_.py
```

To run main.py, use the following command on a new terminal,

```
python /path/to/main.py
```

When ran, will promt to user to input various file paths and environment paths, as follows:

```
Enter the path to the docking environment:
Enter the path for general analysis (Python environment):
Enter the directory path where your scripts are located:
Enter the directory path where you want to save the results:
Enter the path to the fpocket executable:
Enter the paths to the FASTA files for analysis separated by a space:
Enter the paths to the PDB files for fpocket analysis separated by a space:
Enter the paths to the PDB files to convert to PDBQT separated by a space:
Enter the paths to the SDF files to convert to PDBQT separated by a space:
```

To run Docking_.py, make sure you are in your docking environment, and run the command as follows:

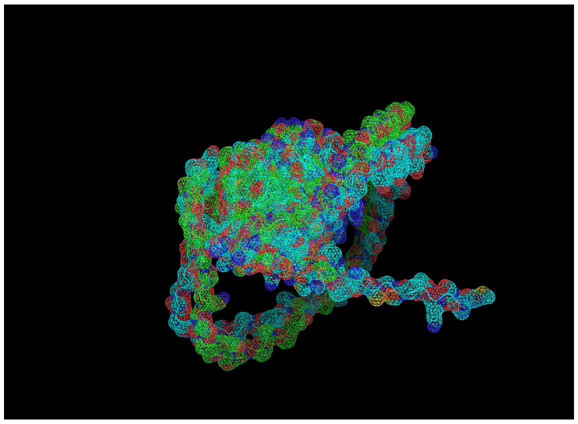
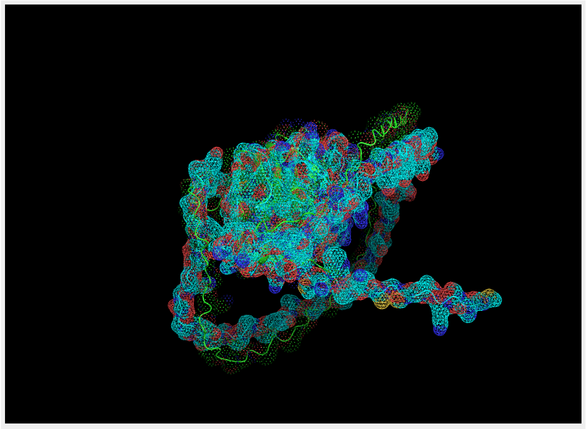
```
python path/to/Docking_.py path/to/receptor/fileisoform_1.pdbqt
path/to/ligand/file/'ValproicAcid.pdbqt' path/to/config.txt
path/to/and/name/of/outpul_file/1+val.pdbqt
```

Getting Started with the Pipeline:

1. **esm_fold.py:**
   On running this script it will generate .pdb files, the files can be located in the path where the script was saved, that can then be transferred/duplicated to your local computer for further analysis.

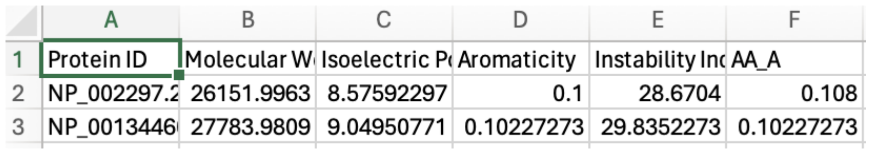
2. **pymol_.py:**
   When this script is run on pymol, it will output multiple .txt file, each of RMSD score, b-factors, and .png file for electrostatic scurface, and hydrophobic surface.

   Sample output:
   RMSD between isoform_1 and isoform_3: 0.8428471684455872 Å
   Average B-factor for isoform_1: 78.90

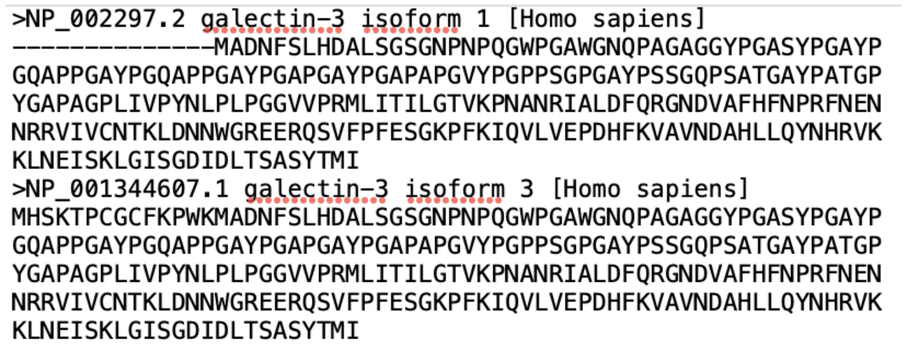
3. **main.py:**
   When main.py is run it will promt the user to input multiple paths for various files, and environments. The first result that main.py produces is of ProtParam, which is present in .csv format, as protein_analysis.csv, in the result directory that the user gave path to. Then it is going to concatenate using concat.py, the multiple .fasta files, that was provided into one, to run the Clustal-Omega, which will produce an aligned_sequences.fasta file, with the multiple sequences aligned. It is then going to run run_fpocket.py, in the fPocket environment, which will be saved as FileName_out folder in the result path. After the user inputs the filepaths for the .pdb and .sdf files to be prepped for autodock, the final result will be saved as .pdbqt file in the result path.

   Sample outputs:
   (a) protein_analysis.csv:



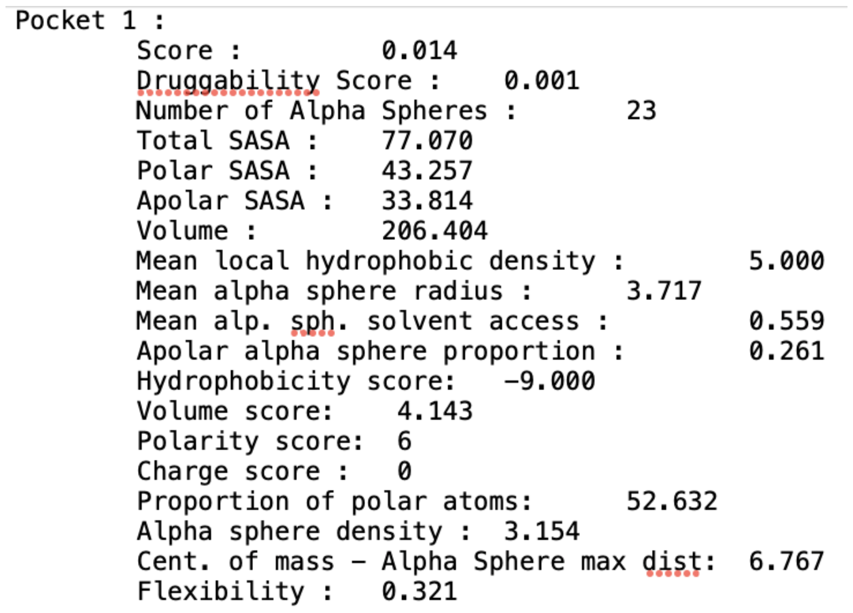
	A	B	C	D	E	F
1	Protein ID	Molecular W	Isoelectric P	Aromaticity	Instability Inc	AA_A
2	NP_002297.2	26151.9963	8.57592297	0.1	28.6704	0.108
3	NP_0013446	27783.9809	9.04950771	0.10227273	29.8352273	0.10227273

(b) aligned_sequences.fasta:



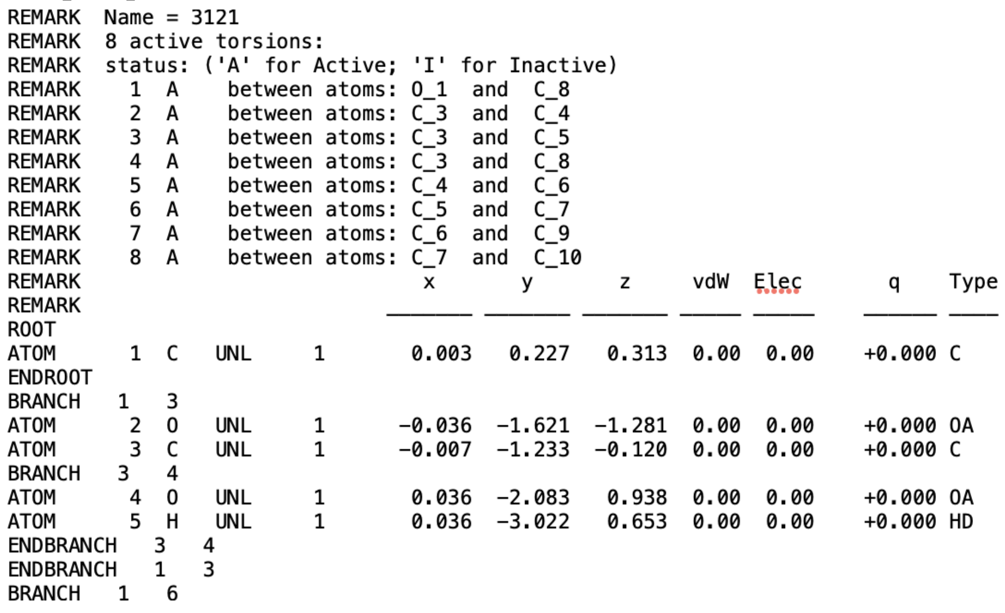
```
>NP_002297.2 galectin-3 isoform 1 [Homo sapiens]
--------------MADNFSLHDALSGSGNPNPQGWPGAWGNQPAGAGGYPGASYPGAYP
GQAPPGAYPGQAPPGAYPGAPGAYPGAPAPGVYPGPPSGPGAYPSSGQPSATGAYPATGP
YGAPAGPLIVPYNLPLPGGVVPRMLITILGTVKPNANRIALDFQRGNDVAFHFNPRFNEN
NRRVIVCNTKLDNNWGREERQSVFPFESGKPFKIQVLVEPDHFKVAVNDAHLLQYNHRVK
KLNEISKLGISGDIDLTSASYTMI
>NP_001344607.1 galectin-3 isoform 3 [Homo sapiens]
MHSKTPCGCFKPWKMADNFSLHDALSGSGNPNPQGWPGAWGNQPAGAGGYPGASYPGAYP
GQAPPGAYPGQAPPGAYPGAPGAYPGAPAPGVYPGPPSGPGAYPSSGQPSATGAYPATGP
YGAPAGPLIVPYNLPLPGGVVPRMLITILGTVKPNANRIALDFQRGNDVAFHFNPRFNEN
NRRVIVCNTKLDNNWGREERQSVFPFESGKPFKIQVLVEPDHFKVAVNDAHLLQYNHRVK
KLNEISKLGISGDIDLTSASYTMI
```

(c) fPockets:



```
Pocket 1 :
        Score :         0.014
        Druggability Score :    0.001
        Number of Alpha Spheres :       23
        Total SASA :    77.070
        Polar SASA :    43.257
        Apolar SASA :   33.814
        Volume :        206.404
        Mean local hydrophobic density :        5.000
        Mean alpha sphere radius :      3.717
        Mean alp. sph. solvent access :         0.559
        Apolar alpha sphere proportion :        0.261
        Hydrophobicity score:   -9.000
        Volume score:    4.143
        Polarity score:  6
        Charge score :   0
        Proportion of polar atoms:      52.632
        Alpha sphere density :  3.154
        Cent. of mass - Alpha Sphere max dist:  6.767
        Flexibility :   0.321
```

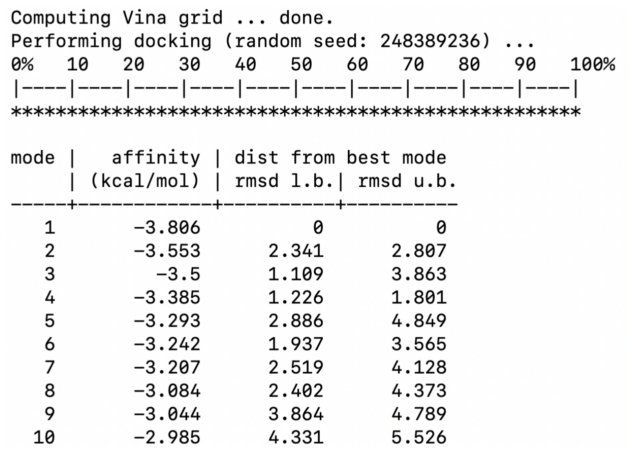
(d) .pdbqt file:



```
REMARK  Name = 3121
REMARK  8 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK    1  A    between atoms: O_1  and  C_8
REMARK    2  A    between atoms: C_3  and  C_4
REMARK    3  A    between atoms: C_3  and  C_5
REMARK    4  A    between atoms: C_3  and  C_8
REMARK    5  A    between atoms: C_4  and  C_6
REMARK    6  A    between atoms: C_5  and  C_7
REMARK    7  A    between atoms: C_6  and  C_9
REMARK    8  A    between atoms: C_7  and  C_10
REMARK                            x       y       z     vdW  Elec       q    Type
REMARK                         _______ _______ _______ _____ _____    ______ ____
ROOT
ATOM      1  C   UNL     1       0.003   0.227   0.313  0.00  0.00    +0.000 C
ENDROOT
BRANCH   1   3
ATOM      2  O   UNL     1      -0.036  -1.621  -1.281  0.00  0.00    +0.000 OA
ATOM      3  C   UNL     1      -0.007  -1.233  -0.120  0.00  0.00    +0.000 C
BRANCH   3   4
ATOM      4  O   UNL     1       0.036  -2.083   0.938  0.00  0.00    +0.000 OA
ATOM      5  H   UNL     1       0.036  -3.022   0.653  0.00  0.00    +0.000 HD
ENDBRANCH   3   4
ENDBRANCH   1   3
BRANCH   1   6
```

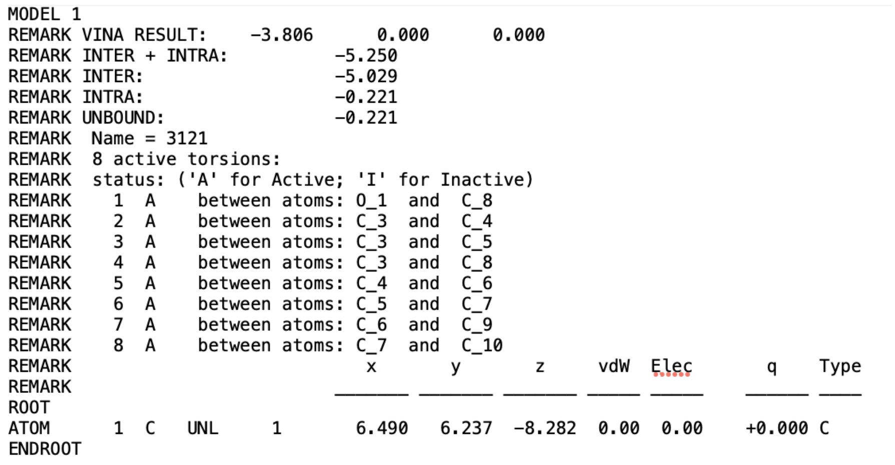
4. **Docking_.py:**
   This script outputs the .pdbqt files to the result path, which contains various binding models, and vina scores, along with printing a summary on the terminal.

   Sample outputs:



```
Computing Vina grid ... done.
Performing docking (random seed: 248389236) ...
0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
***************************************************

mode |   affinity | dist from best mode
     | (kcal/mol) | rmsd l.b.| rmsd u.b.
-----+------------+----------+----------
   1       -3.806          0          0
   2       -3.553      2.341      2.807
   3         -3.5      1.109      3.863
   4       -3.385      1.226      1.801
   5       -3.293      2.886      4.849
   6       -3.242      1.937      3.565
   7       -3.207      2.519      4.128
   8       -3.084      2.402      4.373
   9       -3.044      3.864      4.789
  10       -2.985      4.331      5.526
```



```
MODEL 1
REMARK VINA RESULT:    -3.806      0.000      0.000
REMARK INTER + INTRA:          -5.250
REMARK INTER:                  -5.029
REMARK INTRA:                  -0.221
REMARK UNBOUND:                -0.221
REMARK  Name = 3121
REMARK  8 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK    1  A    between atoms: O_1  and  C_8
REMARK    2  A    between atoms: C_3  and  C_4
REMARK    3  A    between atoms: C_3  and  C_5
REMARK    4  A    between atoms: C_3  and  C_8
REMARK    5  A    between atoms: C_4  and  C_6
REMARK    6  A    between atoms: C_5  and  C_7
REMARK    7  A    between atoms: C_6  and  C_9
REMARK    8  A    between atoms: C_7  and  C_10
REMARK                            x       y       z     vdW  Elec       q    Type
REMARK                         _______ _______ _______ _____ _____    ______ ____
ROOT
ATOM      1  C   UNL     1       6.490   6.237  -8.282  0.00  0.00    +0.000 C
ENDROOT
```

Pitfalls and Limitations

The Protein Analysis and Docking Pipeline, while robust and comprehensive, encompasses several limitations and challenges that users should consider. The success and reliability of the pipeline heavily depend on the proper installation and configuration of multiple external tools, which can be complex and time-consuming. Moreover, it requires significant computational resources, which may not be readily available to all users, potentially limiting its utility in resource-constrained environments.