**Mutation Induced 3D Protein Structure Prediction Manual**

# 

**Table of Contents**

[**Introduction 1**](#_neqp2ep6wbue)

[● About the pipeline 1](#_ib6iw3kwgfwg)

[● Workflow 1](#_1w4ftiurxry4)

[**Packages 2**](#_1znrxa9i7se5)

[● ESMFold 2](#_kasce87x0ycb)

[● Biopython 2](#_tzs1h9sehyhi)

[● DSSP 2](#_4fljj0ma8sr)

[**Package Installation 3**](#_kiqzmcy659ej)

[ESMfold Model Prediction 3](#_1dia7896f5m0)

[1. Bridges2 Workspace 3](#_nxesikitriil)

[2. ESMfold Installation 3](#_c2x8i3wjwi31)

[3. OpenFold Installation 3](#_yk4zl9v7m0kk)

[PDB Model Analysis 4](#_3op1ev7jq7r4)

[1. Download model folder 4](#_8axhgck4m5wb)

[2. Conda Environment setup 4](#_vh3o3vjlntfl)

[**Running Pipeline 5**](#_u7qp4hieqrs5)

[1. Connect to Bridges. 5](#_q4yyfvja8ys)

[2. Model Prediction 5](#_v9gmpayzx0xw)

[3. PDB Analysis 7](#_fv81y5gd9ya9)

[**Limitations 8**](#_8s0e6kesy5qx)

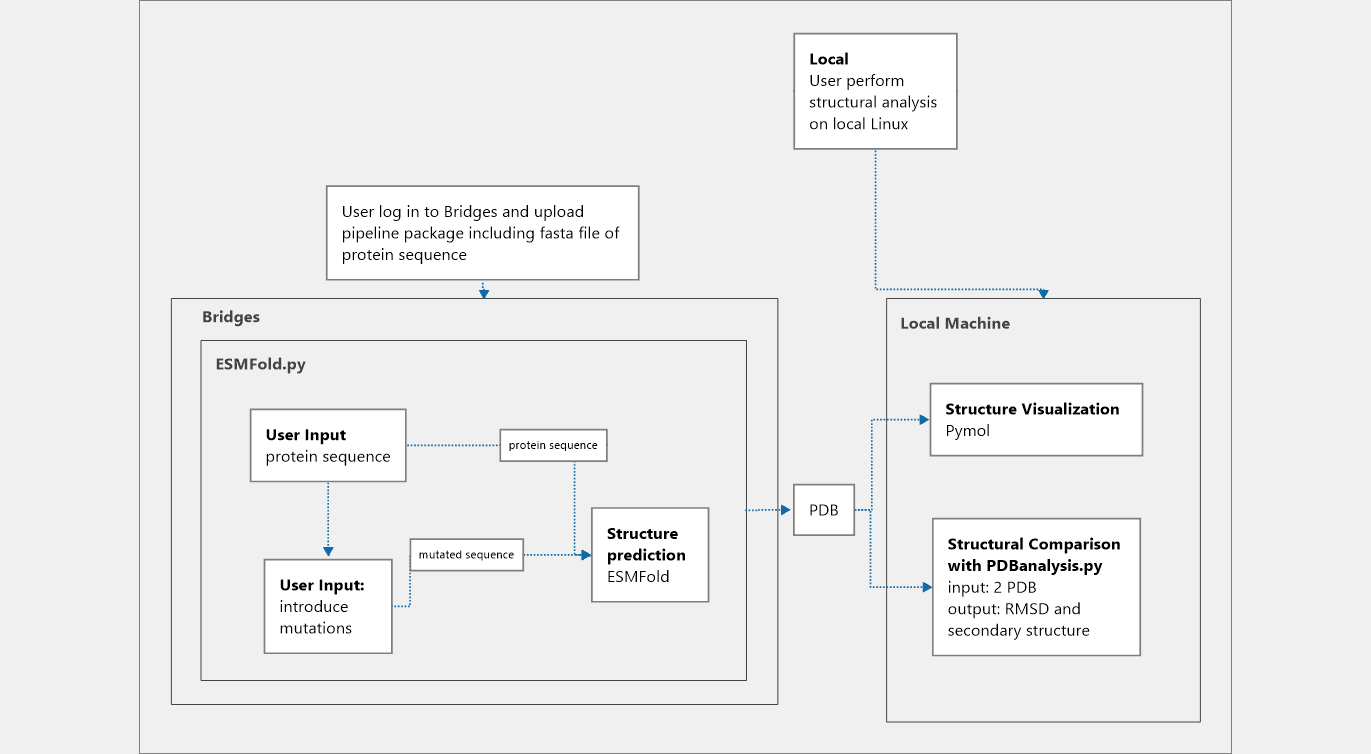
# 

# Introduction

## About the pipeline

This pipeline of monomer 3D structure prediction given protein sequences contains two python scripts, ESMFold.py and PDBanalysis.py. ESMFold.py performs mutations and structure prediction using ESMFold and the PDBanalysis.py performs structural analysis using the Biopython PDB module. In running ESMFold.py, the user will provide a protein sequence they wish to modify and introduce mutations into the sequence as input. The 3D structure of the mutated sequence will be generated from ESMFold prediction, which is a structural prediction tool that generates the protein structure based on the amino acid sequence of interest. After getting the predicted structures, users will have two options to perform structural analysis. One is to continue to the next section of this pipeline and compare two structures using PDBanalysis.py, which will produce the root mean square deviation (RMSD) as the measurement of the difference between the two protein structures. The other option is to visualize secondary protein structures using Pymol, a visualization tool for protein structure.

## Workflow

****

# Packages

## ESMFold

ESMFold is a transformer protein language model from the Meta Fundamental AI Research Protein Team (FAIR). The model allows users to perform structural prediction end to end directly from the sequence of their protein of interest.

Details about the model, source code, and database used can be found in the github repository: <https://github.com/facebookresearch/esm/tree/main>

## Biopython

Biopython is a set of free modules available for computational analysis of biological data in Python. Biopython includes Python libraries and applications that help for processing biological data files commonly encountered in bioinformatics, such as Fasta file and PDB structural file. The libraries can be directly installed as Bio in the conda environment.

Please refer to the following website for Biopython packages:

<https://biopython.org/wiki/Documentation>

In this pipeline specifically, Biopython is used for extracting structural information from PDB files and performing structural alignment for PDB objects.

Please refer to the following link for more detail about the package used:

https://biopython.org/wiki/The\_Biopython\_Structural\_Bioinformatics\_FAQ

<https://biopython.org/docs/1.75/api/Bio.PDB.Superimposer.html>

## DSSP

DSSP (Dictionary of Secondary Structure of Proteins) is a Python package under Biopython that helps in predicting protein secondary structure based on its three-dimensional atomic coordinates. The assignment is based on the H-bonding patterns indicated by the residue coordinates. The assignment is achieved through the *Define Secondary Structure of Proteins* Algorithm defined in the Pascal program in 1983, and the algorithm is currently used as the common method for determining protein secondary structure given a PDB file (such as in Pymol).

Please refer to the following link for the source code of the package:

<https://biopython.org/docs/1.75/api/Bio.PDB.DSSP.html>

<https://github.com/biopython/biopython/blob/master/Bio/PDB/DSSP.py>

# Package Installation

To run the complete pipeline, users have to construct a conda environment compatible with the required packages. However, as the program is designed to run in two parts, the user has to construct two different conda environments on Briges2 Server and at their local environment.

## **ESMfold Model Prediction**

We suggest the user run ESMfold on the Bridges2 server to save local storage space and high efficiency. This requires the installation of both openfold and esmfold on their bridges workspace. Here is the instruction for how to set up the conda environment:

### Bridges2 Workspace

We suggest the user move to their own project workspace in ocean when performing the following steps. The installation and running of the prediction model requires large storage spaces.

### ESMfold Installation

To get all necessary files, clone the following repository:

git clone <https://github.com/facebookresearch/esm.git>

The repository includes a yml file that helps you set up a conda environment with all the required packages. Move into the folder and open the yml file with following command:

conda env create -f environment.yml

Then, you can activate the environment:

conda activate esmfold

A required module needs to be downgraded to be compatible with the current environment. Do this by calling:

conda install -c conda-forge einops=0.6.1

### OpenFold Installation

To have the prediction model working you also have to install OpenFold from github repositories. Run the following command in sequence to get all the files you need.

First, interact with a gup node on Bridges2 and load the required modules.

interact –gpu

module load cuda/11.7.1

If you are following the Installation guideline sequentially, you should have the esmfold conda environment activated. If not, please activate the environment before you move on.

Then, install the following packages in the esmfold environment:

* + pip install "fair-esm[esmfold]"
  + pip install'dllogger@git+https://github.com/NVIDIA/dllogger.git'
  + pip install 'openfold @ git+<https://github.com/aqlaboratory/openfold.git@4b41059694619831a7db195b7e0988fc4ff3a307>'
  + pip install git+<https://github.com/facebookresearch/esm.git>

\*Notice: the above command are all one-line commands

Your environment is ready to go!

Lastly, upload the pipeline folder to your workspace on Bridges, and you should be able to run the Python file with your esmfold conda environment activated.

Please refer to the “Running Pipeline” section for how to operate with the Python file.

## PDB Model Analysis

In the second part of the pipeline, we offer a script for analyzing the predicted pdb files that you received on Bridges2. However, this script can only be run locally due to model incompatibility on Bridges2 and it can **only be operated on a Linux based OS**.

Users have to set up a new conda environment locally. We have prepared a yml file for you to complete the set up instantly.

### Download model folder

After you finish all the predictions you need, all the pdf files are placed in a folder named “ESM\_predictions”. This folder also contains the Python file for the script and the yml file that helps you to set up the environment. Please download the folder to your local environment.

### Conda Environment setup

To set up the conda environment, open the terminal and move inside the folder. Create a new conda environment (default name as PDBAnalysis) using the following command:

conda env create -f PDBAnalysis\_[your\_system\_version].yml

Then, activate the new environment:

conda activate PDBAnalysis

With the environment activated, you should be all set to run the script.

You can also choose to visualize individual PDB file if you need further analysis. Pymol is a good tool for doing this. Please refer to the following link if you need to install Pymol locally:

<https://pymol.org/support.html?#installation>

# Running Pipeline

The pipeline is separated into two parts: predicting the protein structure provided specific mutations and analyzing PDB outputs from the prediction. You are welcomed to run those two parts separately based on your need, but here we will walk through how to run the complete pipeline in sequence.

## Connect to Bridges.

Model prediction is expected to operate on Bridges2. To login into Bridges2, input the following command in your terminal:

ssh <username>@bridges2.psc.edu

Bridges2 will ask for your password. After login, move to the corresponding working directory where you upload the pipeline folder (named “Mutation\_induced\_structural\_prediction”)

When login to Bridges2, your default node is a Login node, which is for administrative purposes and does not support heavy calculations. To run actual tasks, you need to connect to one of the computation nodes:

interact –gpu

Please make sure to do this every time you connect to Briges2!

## Model Prediction

The pipeline implementing user proposed mutation in existing fasta files. Before you start the program, please place the fasta file you want to work on in the Pipeline folder.

Then, activate the conda environment and run the script, no extra variables are needed as you will be asked to deliver all parameters when the program runs:

conda activate esmfold

python ESMfold.py

After initiating the program, you will be prompted to input the location of the interested fasta file.

Enter fasta file path:

If the path is not valid, you will encounter an error message that asks you to re-enter the path to achieve the correct file. Once the target fasta file is opened, you will be asked to input the single point mutation you wish to introduce to the sequence in the following formats:

Enter mutations: (if inputting multiple mutations, separate each by space)

If you know the original amino acid at the location:

V10H

Where the first amino acid is the original one in the sequence, the number is the location of the target amino acid, and the last character is the mutation you want to make.

If you do not know the original amino acid in the sequence, you can input the mutation in the following format which indicates the location and the mutation you want to make.

10H

If you wish to introduce multiple mutation at the same time, you can do it by separating the mutations with a space in between them:

V10H 30P A45K

The program will output the mutations that were successfully performed on the input sequence and the mutated sequence to the terminal. Then you will be asked whether you want to save the mutated sequence into a fasta file.

Save sequence to fasta file: [y/n]

If you choose to save the sequence to a new file, the new fasta file will be saved to the same directory as the input fasta file under the name of (input sequence ID)\_(sequence of mutations made)\*.

\* Notice: saving the sequence to a new file is recommended so ESMFold can be run correctly.

Next, you will have the option to process another fasta file:

DO you need to process another fasta file? [y/n]

If you enter yes, you will be asked to enter information regarding the next fasta file you want to process. If you are done with modifying the amino acid sequence, you will move on to the model prediction section using ESMFold.

The next section of the script loads a pretrained model, and computes the 3D structure of the mutated protein sequence from the previous section using the saved fasta file.

After the prediction finish, you will be prompted to name the pdb file:

Please enter the file name you want to save as, filename should end with .pdb

The resulting pdb file will be in the same directory as the input fasta file.

## PDB Analysis

To run analysis on existing PDB files, set your working directory at the downloaded folder. The folder should have all models you received from the ESMfold prediction. Add PDB files within the folder if you want to perform additional analysis.

Activate the conda environment:

conda activate PDBAnalysis

Then, run the script:

python PDBAnalysis.py

The script will perform pairwise analysis on two PDB files of your choice. The reference model is usually your original protein without mutation, and the sample model is the predicted structure of the protein carrying the proposed mutation. Input the PDB file name as being requested by the script, remember to include the .pdb suffix.

The RMSD of the two models will be printed out directly in the terminal. For the secondary structure prediction result, you can either [1] save it to a local csv file for data analysis OR [2] print it out in the terminal. Choose the option you prefer by typing the corresponding number in the terminal.

You can refer to the following chart for DSSP codes of secondary structure

===== ====

Code Structure

===== ====

H Alpha helix (4-12)

B Isolated beta-bridge residue

E Strand

G 3-10 helix

I Pi helix

T Turn

S Bend

\- None

===== ====

# Limitations

There are a few limitations in our pipeline. One of the limitations is the availability of types of mutations the user can perform on the amino acid sequence. Our pipeline only allows single point mutations and does not allow insertion and deletion since the comparison of protein structures is only accurate when the sequences have the same length.

Another limitation is that the user has to create two separate conda environments in order to successfully run the pipeline. Modules used for PDB structural analysis require a specific libboost module version that is not compatible with the environmental setup on Bridges. Thus we are not able to have all the sections of the pipeline set up on Bridges2 and users will have to work in both remote and local environments to walk through the complete pipeline. However, there is also the option that the user might want to perform specific portion of the pipeline and thus will only have to work on one of the two environments.