The phi angle is the dihedral angle formed between the Cα-N bonds through interactions between 2 conjoining amino acids. The psi angle is the dihedral angle formed between the Cα-C bond through interactions of the N-C-Ca-N atoms between 2 conjoining amino acids.

To acquire dihedral angles from your PDB file, you can use biopython. For this example, the PDB file for protein 4N6N was used. You can download the PDB file for your protein of interest on the RCSB PDB website: https://www.rcsb.org/structure/4N6N

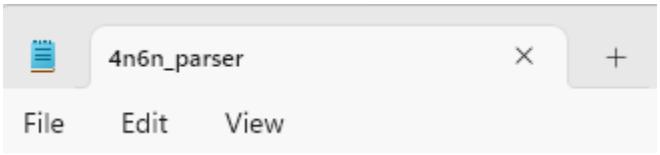
You will need Biopython to access PDBParser to extract the necessary data from your PDB file. For the pdb_file_path, you can replace the example file with the path to your own PDB file. PDBParser module source found here: https://warwick.ac.uk/fac/sci/moac/people/students/peter_cock/python/ramachandran/calculate/#BioPython (Note: this code was edited to include a file path)

```python
import Bio.PDB
from Bio.PDB import PDBParser

#PDB file reader

def main():
    pdb_file_path = r"C:\Users\Tania\Documents\Bioinformatics\Protein Bioinformatics\4n6n.pdb"
    parser = PDBParser
    structure = parser.get_structure('protein', pdb_file_path)

for model in Bio.PDB.PDBParser().get_structure("4N6N", r"C:\Users\Tania\Documents\Bioinformatics\Protein Bioinformatics\4n6n.pdb") :
    for chain in model :
        polypeptides = Bio.PDB.PPBuilder().build_peptides(chain)
        for poly_index, poly in enumerate(polypeptides) :
            print ("Model %s Chain %s" % (str(model.id), str(chain.id))),
            print ("(part %i of %i)" % (poly_index+1, len(polypeptides))),
            print ("length %i" % (len(poly))),
            print ("from %s%i" % (poly[0].resname, poly[0].id[1])),
            print ("to %s%i" % (poly[-1].resname, poly[-1].id[1]))
            phi_psi = poly.get_phi_psi_list()
            for res_index, residue in enumerate(poly) :
                res_name = "%s%i" % (residue.resname, residue.id[1])
                print (res_name, phi_psi[res_index])
```

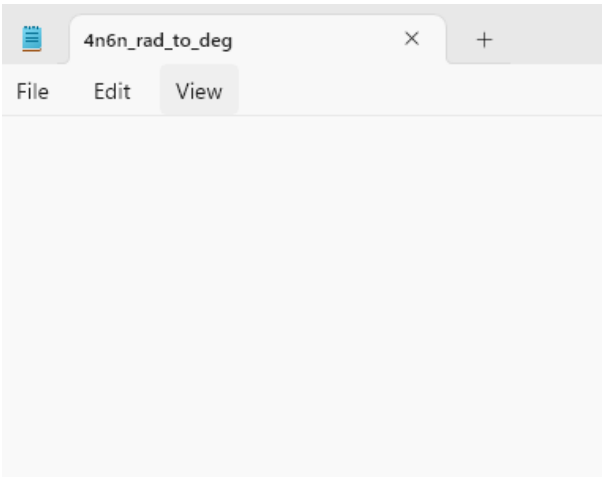
After inputting your PDB file into the program “reader_PDB.py”, you will then receive an output in the program’s terminal. You can then copy and paste the output onto a notepad file:

```
Model 0 Chain A
(part 1 of 3)
length 121
from GLY26
to THR146
GLY26 (None, 1.0759131600635423)
GLY27 (2.8422732602553893, 2.9856141150538598)
LYS28 (-1.9517648336615823, -0.6906251709242297)
HIS29 (-1.9302022284973064, 2.2007684160610483)
TRP30 (-2.0732327665778563, 2.650215320451784)
VAL31 (-2.4162259146490466, 2.395793416914984)
VAL32 (-2.1330254415311902, 2.1731421536833757)
ILE33 (-2.0407257577051925, 2.2581105674424204)
VAL34 (-2.3061751528296757, 2.153831893010605)
ALA35 (-1.7729260469564396, 2.1494923222449174)
GLY36 (-1.6199425540794166, -0.03903658630951815)
SER37 (-2.1145335192433343, 3.015055398716476)
ASN38 (-2.527169761807301, 3.1260876121695396)
GLY39 (1.5942406541362533, 2.591742037086957 5)
TRP40 (-1.0654963052734547, -0.6762101666031828)
TYR41 (-1.2977216694369518, -0.1639781628660284)
ASN42 (-1.737149536985012, 0.2872948825757097)
TYR43 (-0.5586304277469437, -0.9048439666303671)
ARG44 (-0.9044543516813437, -0.6351005050579855)
HIS45 (-1.226539914117705, -0.6952058629868663)
GLN46 (-1.3281735185430106, -0.5627384897993521)
```

The output of the PDBParser will come out as radians. Save this file as a .txt input file. You will need this input for interpretation by the next module. See the input .txt file example:




For the next step, create a .txt output. This file will receive data output. Copy the path of this file, then replace the “output_file_path” of the next 2 coding packages with this file.




Based on what you’ve titled your .txt files, they should look like this:

```
input_file_path = r"C:\Users\Tania\Documents\4n6n_parser.txt" #replace your input file path with where you saved the radian coordinates data in your PC
output_file_path = r"C:\Users\Tania\Documents\4n6n_rad_to_deg.txt" #replace your output file path with a .txt document in which you want to save your degree conversions in your PC
convert_rads_to_degs(input_file_path, output_file_path)
```

Add your files to the “radians_to_degrees.py” module below:

```python
import numpy as np
import re

#Radiants to degrees coordinate converter

def convert_rads_to_degs(input_file_path, output_file_path):
    with open(input_file_path, 'r') as input_file:
        lines = input_file.readlines()

    convert_coords = []

    for line in lines:
        parts = line.strip().split(' ')
        residue_name = parts[0]
        radian_coords = parts[1:]

        converted_coords = []
        for coord in radian_coords:
            if coord.lower() == 'none':
                converted_coords.append('None')
            else:
                real_coord = re.sub(r'[^-.\d]+', '', coord)
                try:
                    converted_coords.append(str(np.degrees(float(real_coord))))
                except ValueError:
                    converted_coords.append('None')

        converted_line = ' '.join([residue_name] + converted_coords)

        convert_coords.append(converted_line)

    with open(output_file_path, 'w') as output_file:
        output_file.write('\n'.join(convert_coords))

input_file_path = r"C:\Users\Tania\Documents\4n6n_parser.txt" #replace your input file path with where you saved the radian coordinates data in your PC
output_file_path = r"C:\Users\Tania\Documents\4n6n_rad_to_deg.txt" #replace your output file path with a .txt document in which you want to save your degree conversions in your PC
convert_rads_to_degs(input_file_path, output_file_path)
```

After inputting your file from the previous program, you will receive an output file like this: