

Biomolecular Structures A3

Raunak Seksaria : 2023113019

8th February, 2025

1 Question 1

1.1 Alpha Helices:

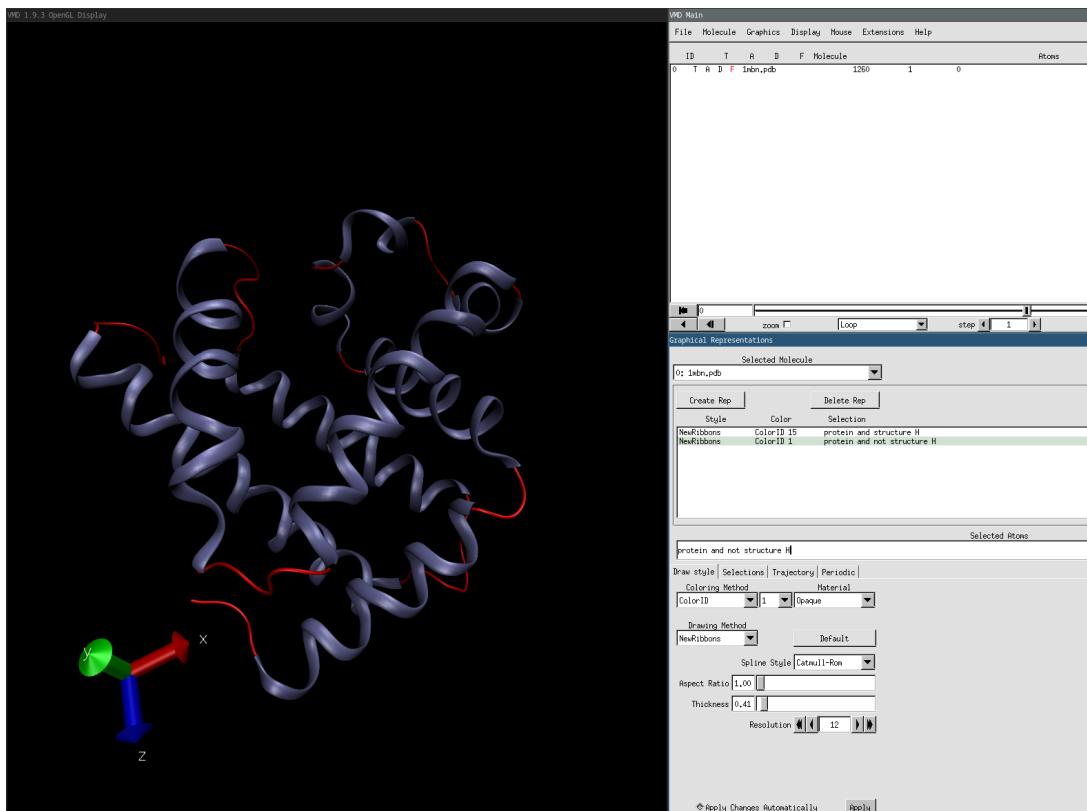


Figure 1: Selected Alpha Helix in 1mbn.pdb

Here is code that I ran to separate the helices and display their resid numbers:

```
set all_helices [atomselect top "protein and structure H"]
set resid_list [lsort -integer -unique [$all_helices get resid]]

set prev_resid -999
set helix_num 1
set helix_residues {}

foreach resid $resid_list {
    if { $prev_resid != -999 && $resid != $prev_resid + 1 } {
        puts "Helix $helix_num: $helix_residues"
        set helix_num [expr $helix_num + 1]
        set helix_residues ""
    }
    append helix_residues "$resid "
```

```

    set prev_resid $resid
}
puts "Helix $helix_num: $helix_residues"

```

The output was:

```

Helix 1: 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Helix 2: 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
Helix 3: 37 38 39 40 41 42
Helix 4: 44 45 46 47 48
Helix 5: 52 53 54 55 56 57
Helix 6: 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
Helix 7: 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96
Helix 8: 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117
118
Helix 9: 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141
142 143 144 145 146 147 148 149

```

Then I displayed 2 of these helices in a different format, for better visualisation.

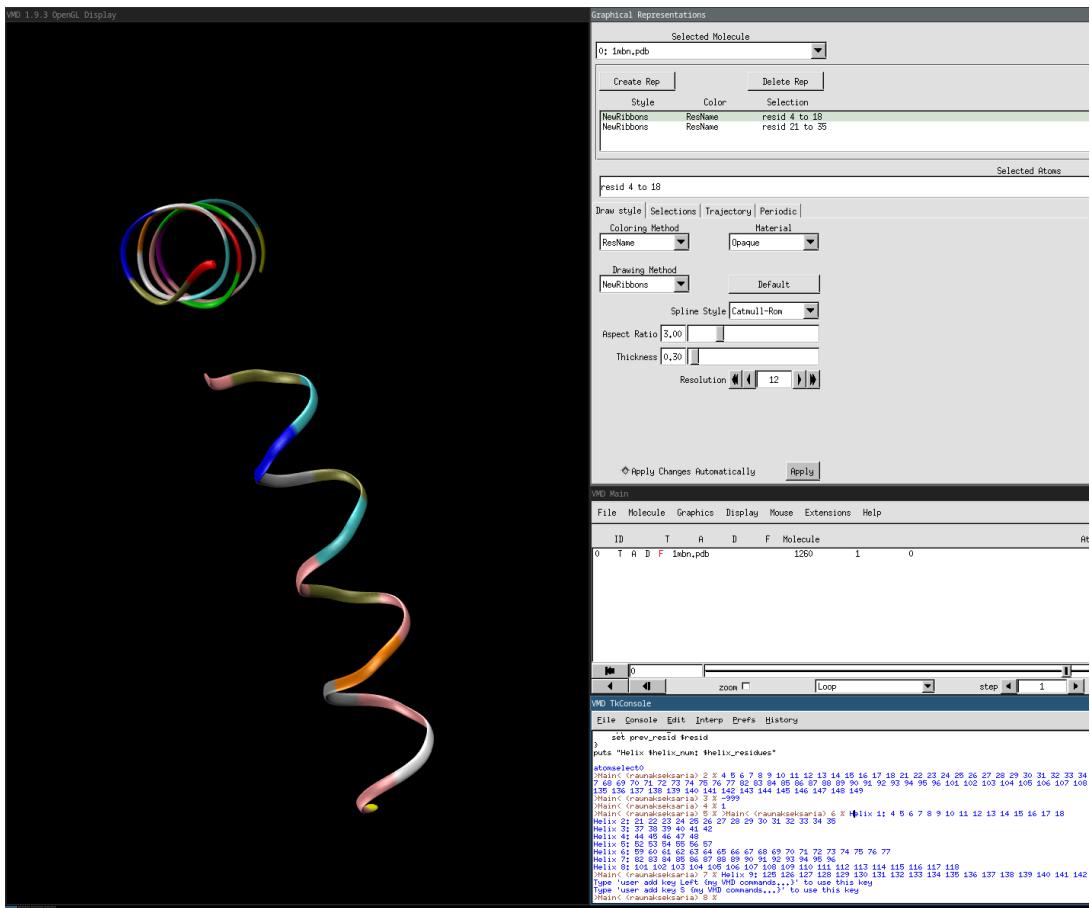


Figure 2: Selected 2 Alpha Helices in different views in 1mbn.pdb

1.2 3-10 Helices:

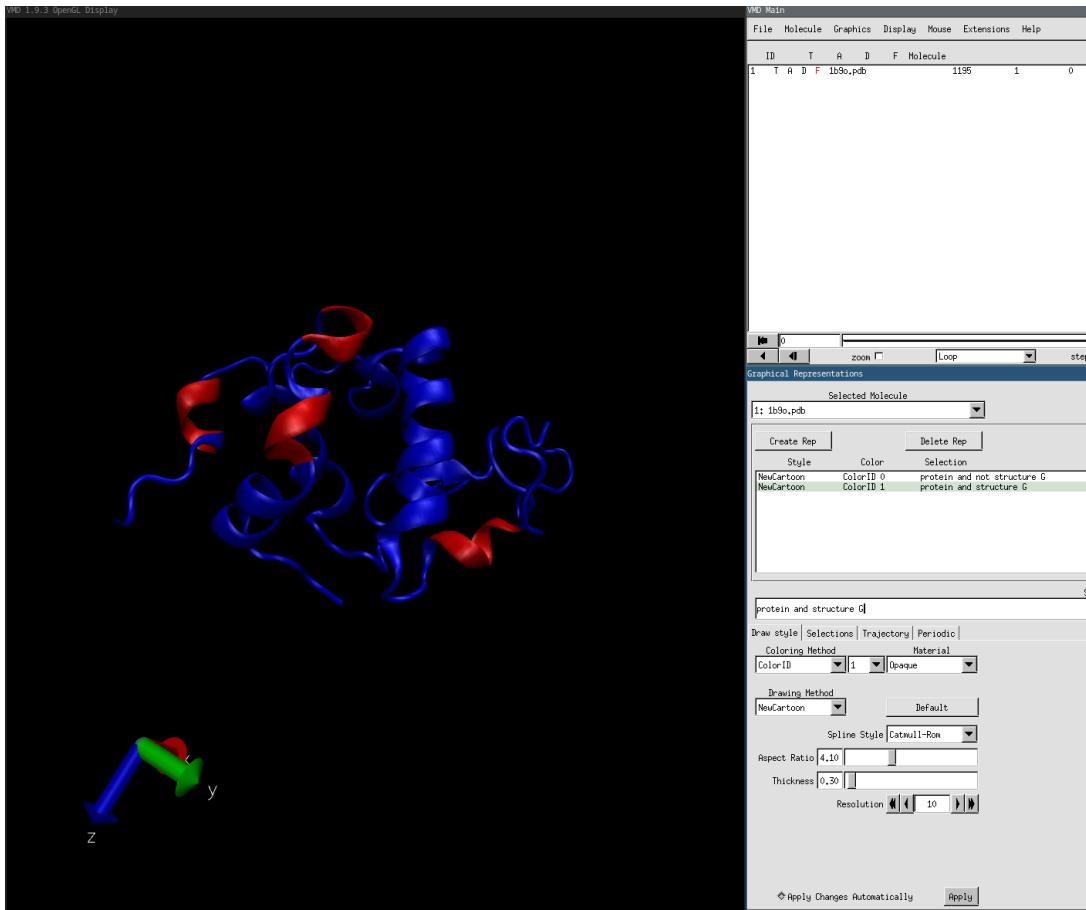


Figure 3: Selected all 3-10 Helices in 1b90.pdb

The input code for separating and extracting the 3-10 helices.

```
# Select all 3-10 helices in the protein
set helix_310 [atomselect top "protein and structure G"]
set resid_list [lsort -integer -unique [$helix_310 get resid]]

# Initialize variables
set prev_resid -999
set helix_num 1
set helix_residues {}

# Loop through residues and group consecutive ones into separate helices
foreach resid $resid_list {
    if { $prev_resid != -999 && $resid != $prev_resid + 1 } {
        puts "3-10 Helix $helix_num: $helix_residues"
        set helix_num [expr $helix_num + 1]
        set helix_residues ""
    }
    append helix_residues "$resid "
    set prev_resid $resid
}
puts "3-10 Helix $helix_num: $helix_residues"
```

The output:

```
3-10 Helix 1: 13 14 15
3-10 Helix 2: 18 19 20
3-10 Helix 3: 77 78 79 80
```

| 3-10 Helix 4: 115 116 117 118

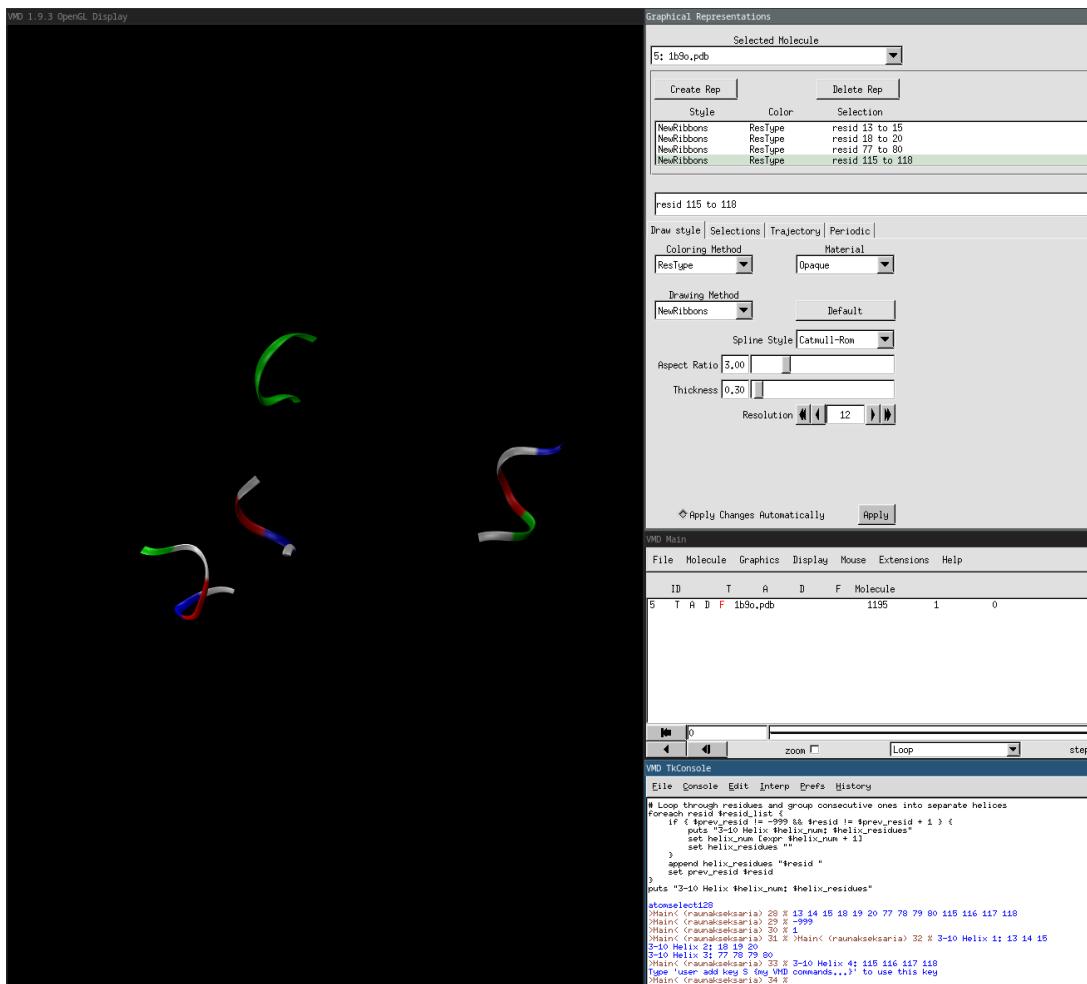


Figure 4: Separated 3-10 helices, we also notice from the figure that length of these 3-10 helices are typically 3-4 residues only. We also notice that similar residues form the helix, ie, only few fixed residues are capable of forming 3-10 helices

1.3 Pi-helix:

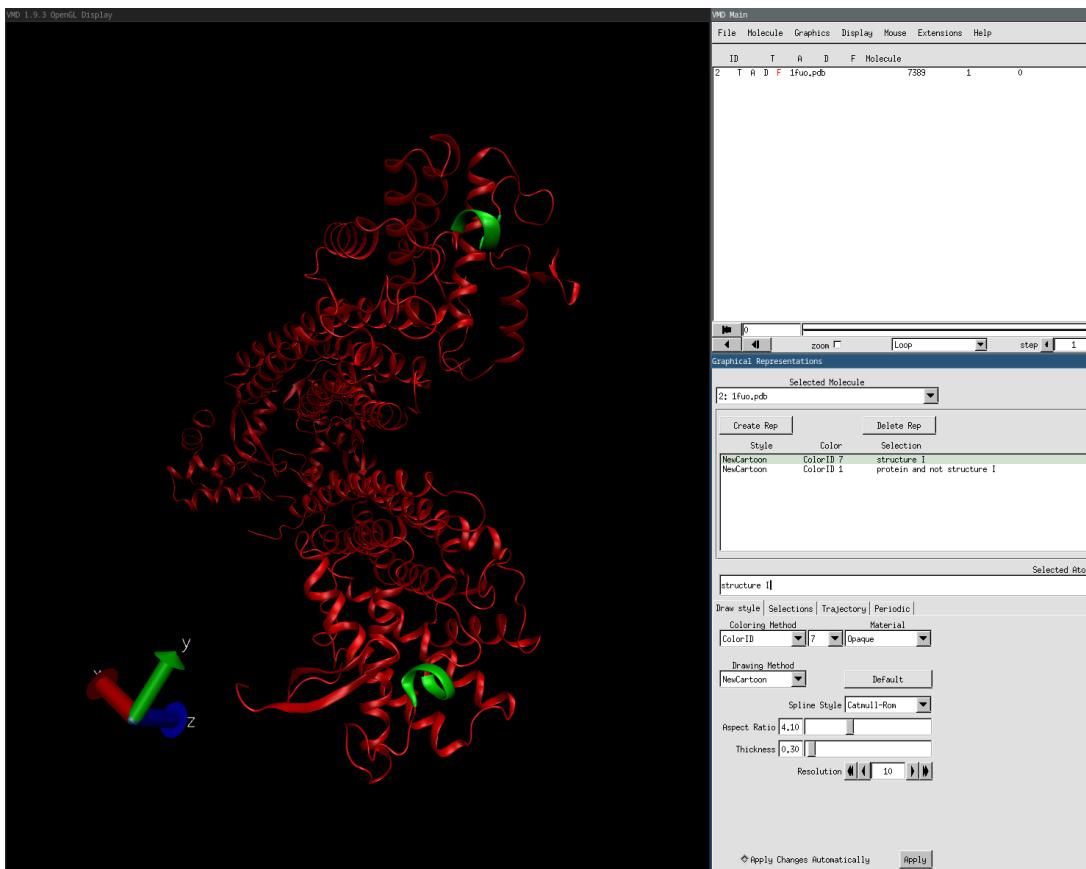


Figure 5: Selected all pi-helix in 1fuo.pdb

```
# Select all pi-helices in the protein
set helix_pi [atomselect top "protein and structure I"]
set resid_list [lsort -integer -unique [$helix_pi get resid]]

# Initialize variables
set prev_resid -999
set helix_num 1
set helix_residues {}

# Loop through residues and group consecutive ones into separate helices
foreach resid $resid_list {
    if { $prev_resid != -999 && $resid != $prev_resid + 1 } {
        puts "Pi-Helix $helix_num: $helix_residues"
        set helix_num [expr $helix_num + 1]
        set helix_residues ""
    }
    append helix_residues "$resid "
    set prev_resid $resid
}
puts "Pi-Helix $helix_num: $helix_residues"
```

Output:

```
Pi-Helix 1: 130 131 132 133 134
```

1.4 Left-handed Helix:

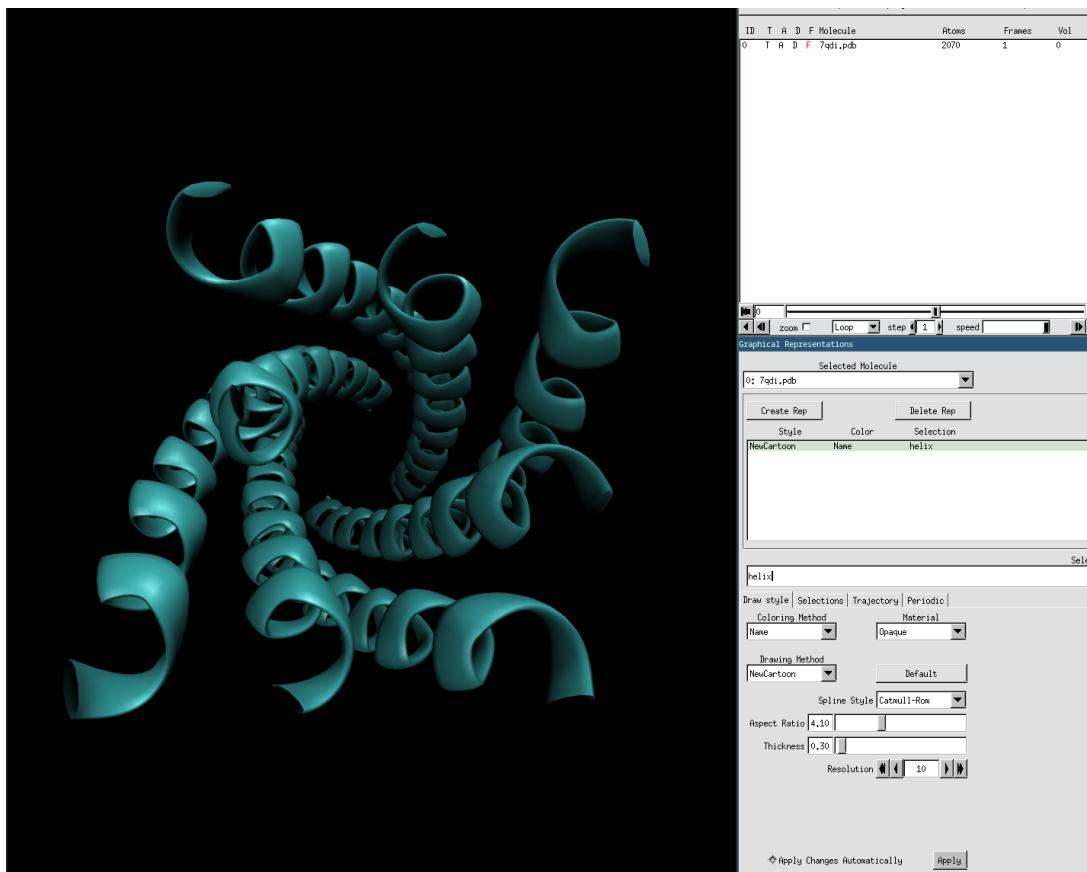


Figure 6: Shown left-handed-helices in 7QDI.pdb

To verify whether these are actually left-handed helices, the Ramachandran plot was obtained, and the angles of psi and phi noted. As expected, the points of the helix were typically in the upper right quadrant.

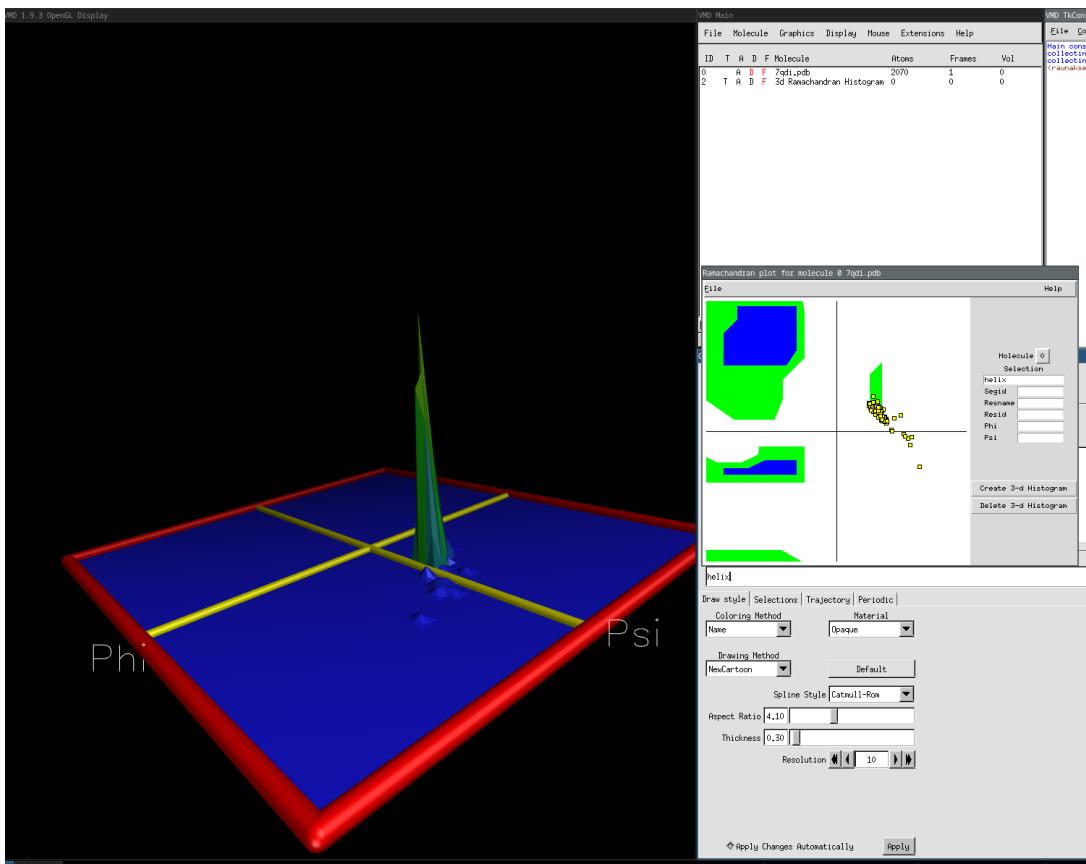


Figure 7: Ramachandran plot and its 3d histogram for 7QDI.pdb

To separate this into individual helices, ran the same code as the one for 3-10 helix. The output was:

```
3-10 Helix 1: 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
      27 28 29
```

1.5 Anti-Parallel Beta Sheet:

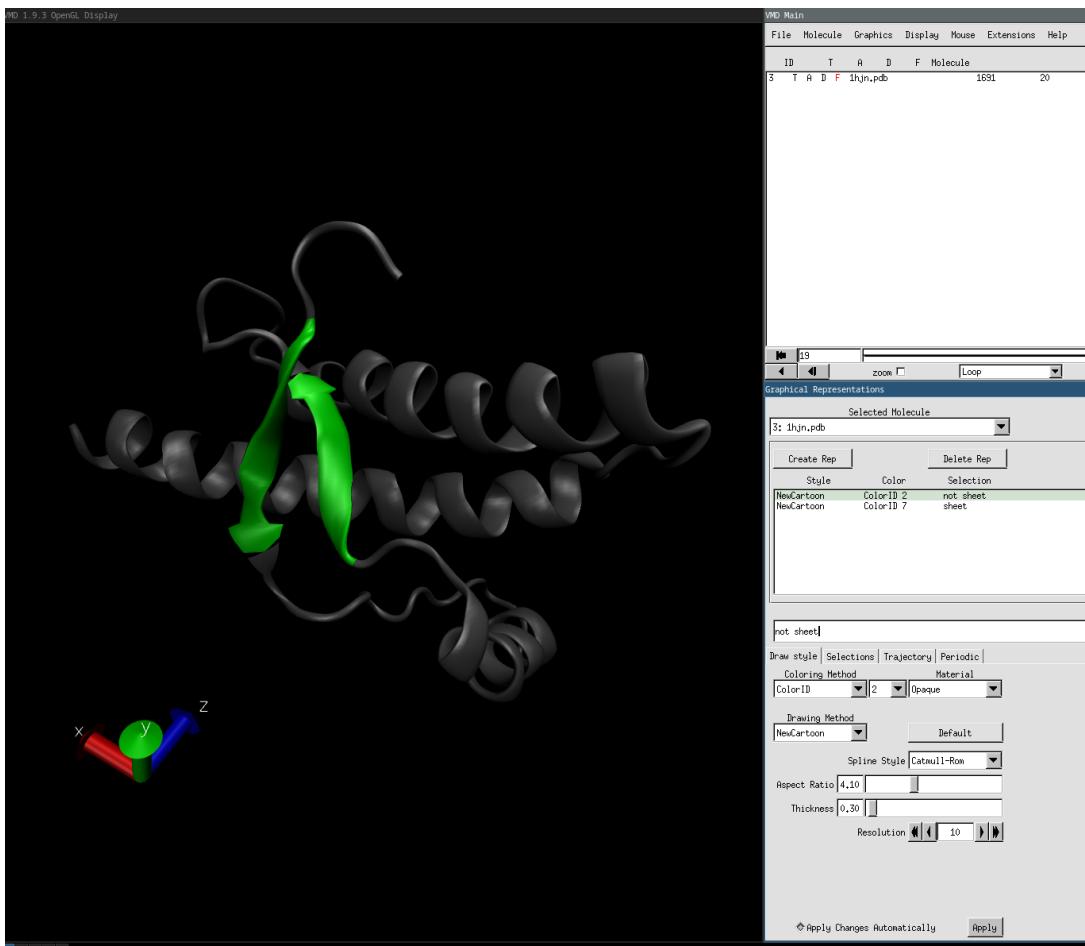


Figure 8: Selected all beta-sheets in 1hjn.pdb

This has only an anti-parallel beta sheet, which was manually verified.

```
# Select beta-sheet residues
set beta_residues [atomselect top "protein and structure E"]
set resid_list [lsort -integer -unique [$beta_residues get resid]]

# Initialize variables
set strand_id 0
set last_resid -999
set strand_residues {}

# Group consecutive residues into strands
foreach resid $resid_list {
    if {[expr $resid - $last_resid] > 2} {
        # New strand detected, assign color and add rep
        if {[llength $strand_residues] > 0} {
            set sel [atomselect top "resid [join $strand_residues]"]
            mol selection "resid [join $strand_residues]"
            mol material Opaque
            mol color ColorID $strand_id
            mol addrep top
            incr strand_id
            if {$strand_id > 32} {set strand_id 0} ;# Reset color after 32
        }
        set strand_residues {} ;# Reset list for new strand
    }
}
```

```

    lappend strand_residues $resid
    set last_resid $resid
}

# Handle the last strand
if {[llength $strand_residues] > 0} {
    set sel [atomselect top "resid [join $strand_residues]"]
    mol selection "resid [join $strand_residues]"
    mol material Opaque
    mol color ColorID $strand_id
    mol addrep top
}

```

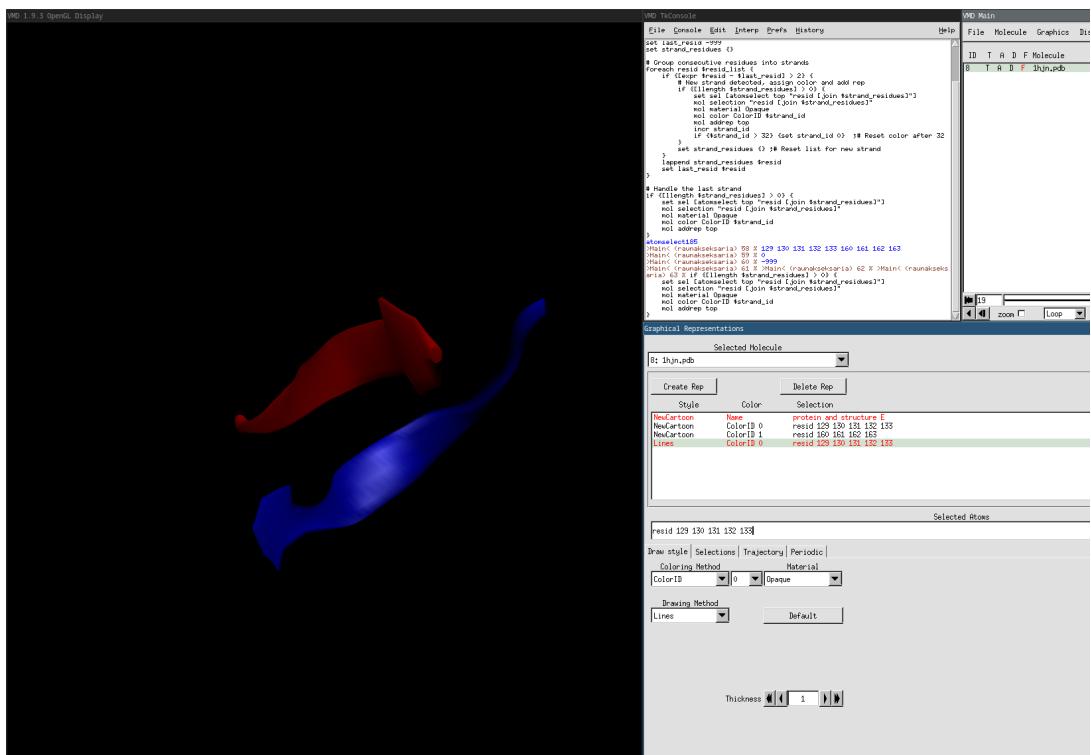


Figure 9: Color coded output after running the previous code on 1HJN.pdb

1.6 Parallel Beta Sheet

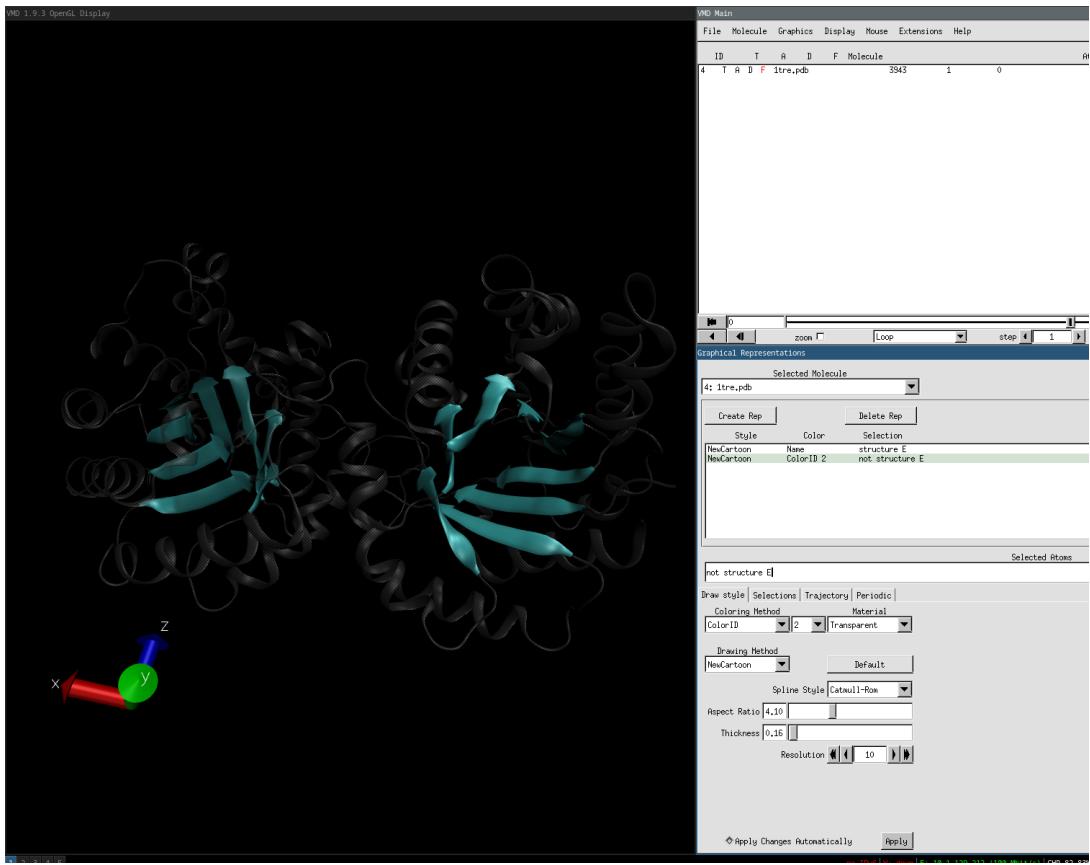


Figure 10: Selected all beta-sheets in 1TRE.pdb

This has only parallel beta sheets, which was manually verified.

The following code is the same as in anti-parallel section.

```
# Select beta-sheet residues
set beta_residues [atomselect top "protein and structure E"]
set resid_list [lsort -integer -unique [$beta_residues get resid]]

# Initialize variables
set strand_id 0
set last_resid -999
set strand_residues {}

# Group consecutive residues into strands
foreach resid $resid_list {
    if {[expr $resid - $last_resid] > 2} {
        # New strand detected, assign color and add rep
        if {[llength $strand_residues] > 0} {
            set sel [atomselect top "resid [join $strand_residues]"]
            mol selection "resid [join $strand_residues]"
            mol material Opaque
            mol color ColorID $strand_id
            mol addrep top
            incr strand_id
            if {$strand_id > 32} {set strand_id 0} ;# Reset color after 32
        }
        set strand_residues {} ;# Reset list for new strand
    }
    lappend strand_residues $resid
    set last_resid $resid
}
```

```

}

# Handle the last strand
if {[llength $strand_residues] > 0} {
    set sel [atomselect top "resid [join $strand_residues]"]
    mol selection "resid [join $strand_residues]"
    mol material Opaque
    mol color ColorID $strand_id
    mol addrep top
}

```

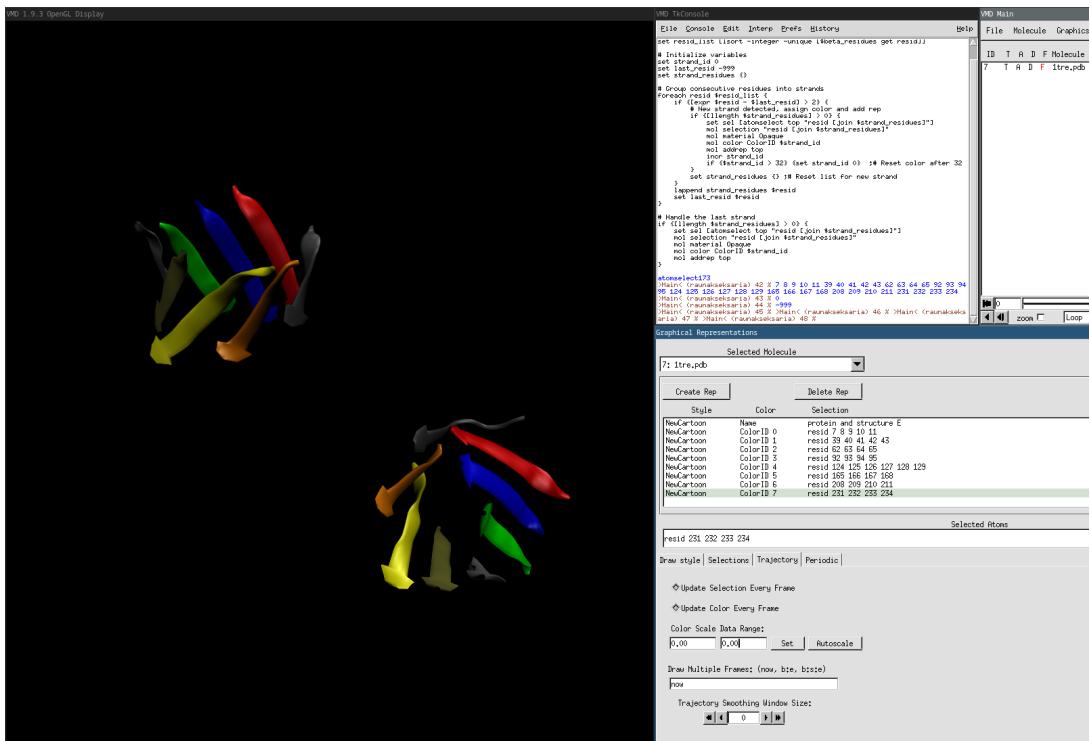


Figure 11: Color coded output after running the previous code on 1HJN.pdb

1.7 Turn with Glycine

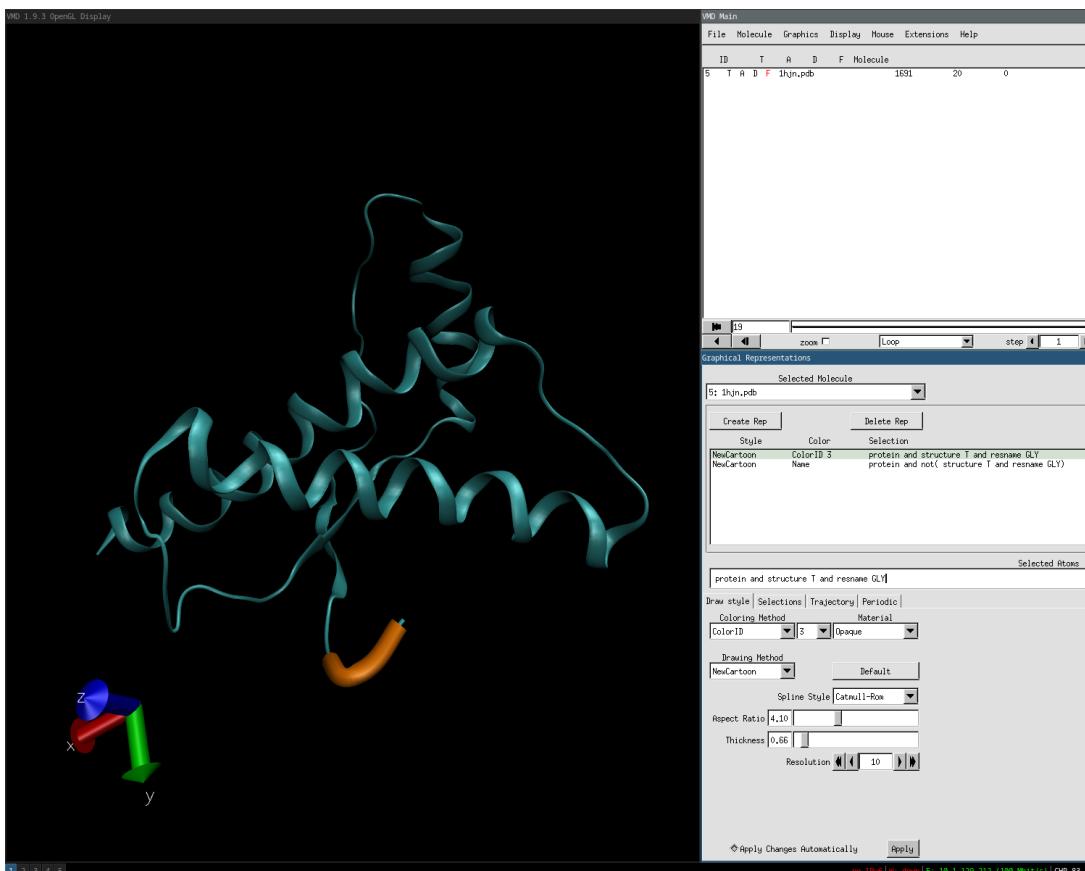


Figure 12: GLY turn on 1HJN.pdb

```
set glyturn [atomselect top "protein and structure T and resname GLY"]
put [lsort -unique [$glyturn get resid]]
```

Output:

```
126 127
```

1.8 Turn with Proline

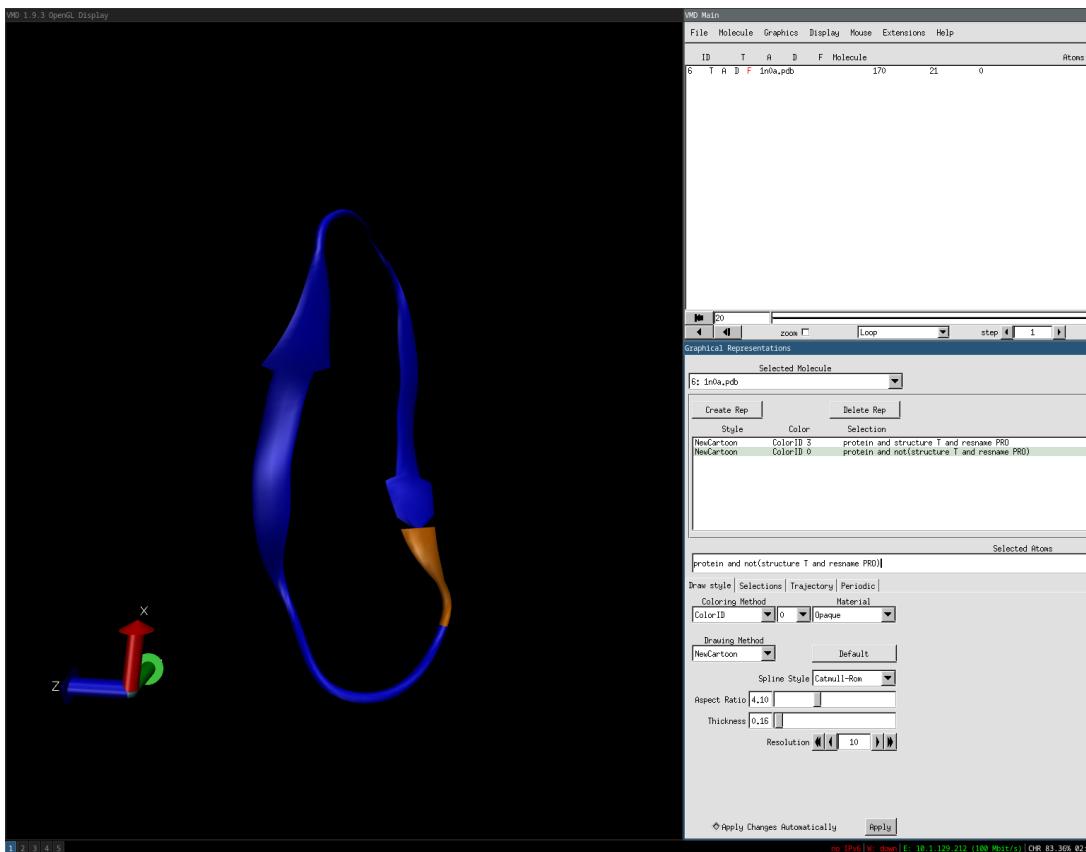


Figure 13: PRO turn in 1n0a.pdb

```
set protturn [atomselect top "protein and structure T and resname PRO"]
put [lsort -unique [$protturn get resid]]
```

Output:

```
5
```

2 Question 2

2.1 Hydrogen bonds

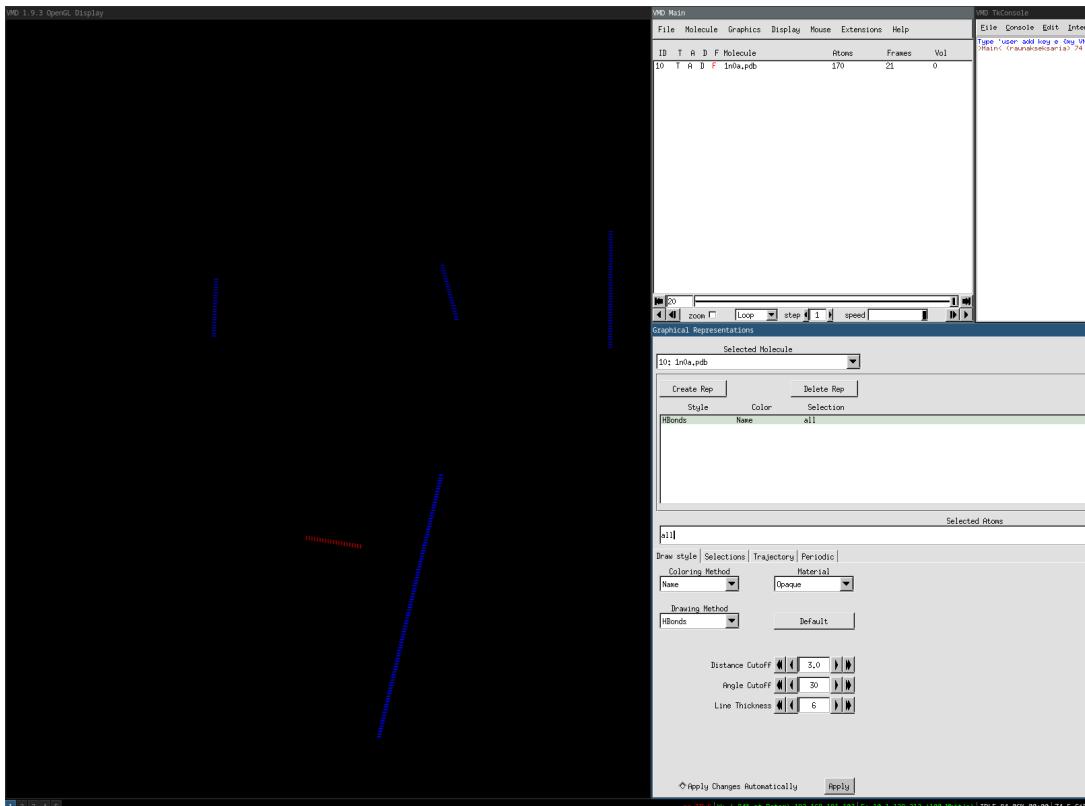


Figure 14: Hydrogen bonds in 1n0a.pdb

```
# Define hydrogen bond cutoff values
set dist_cutoff 3.5      ;# Max donor-acceptor distance (Angstrom)
set angle_cutoff 30       ;# Max hydrogen bond angle (degrees)

# Select all protein atoms
set all_protein [atomselect top "protein"]

# Compute hydrogen bonds
set hbonds [measure hbonds $dist_cutoff $angle_cutoff $all_protein]

# Extract donor and acceptor atom indices
set donors [lindex $hbonds 0]
set acceptors [lindex $hbonds 1]

# Collect unique residue IDs involved in H-bonds
set hb_residues {}

foreach idx $donors {
    set sel [atomselect top "index $idx"]
    lappend hb_residues [$sel get resid]
    $sel delete
}
foreach idx $acceptors {
    set sel [atomselect top "index $idx"]
    lappend hb_residues [$sel get resid]
    $sel delete
}
```

```

# Remove duplicate residues
set hb_residues [lsort -integer -unique $hb_residues]

# Color residues uniquely
set color_id 0
foreach resid $hb_residues {
    set sel [atomselect top "resid $resid"]
    mol selection "resid $resid"
    mol material Opaque
    mol color ColorID $color_id
    mol addrep top
    incr color_id
    if {$color_id > 32} {set color_id 0} ;# Reset color after 32 colors
    $sel delete
}

# Cleanup
$all_protein delete

```

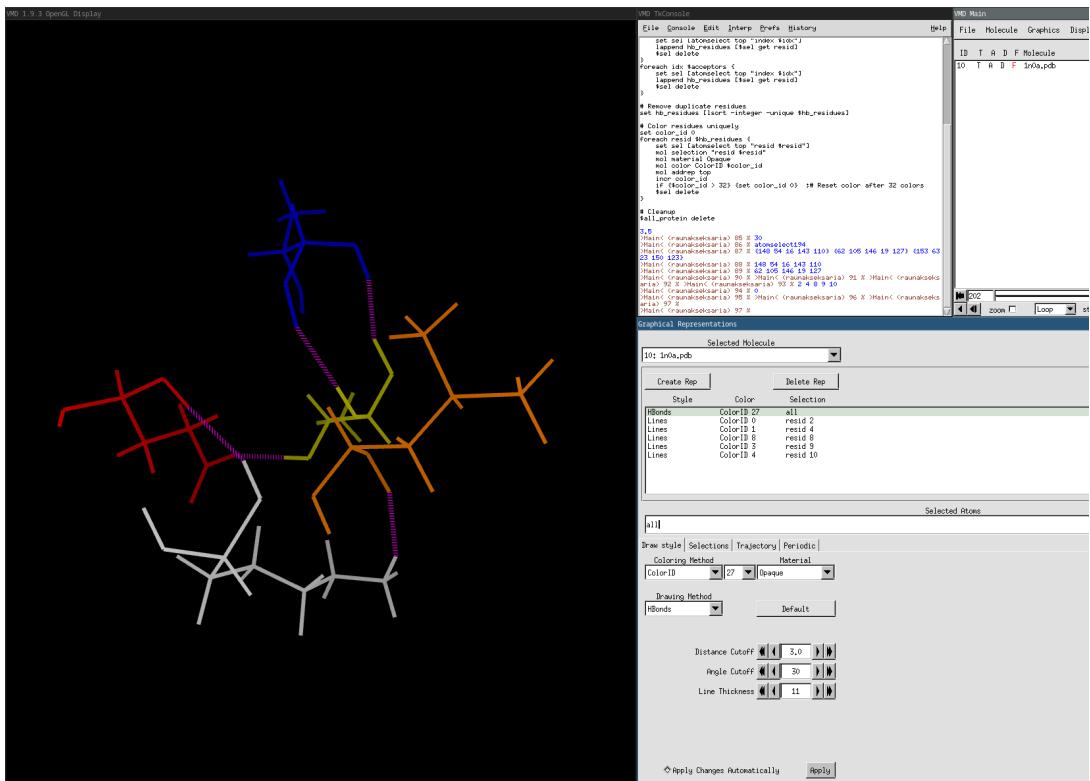


Figure 15: Hydrogen bonds with their related residues color coded in 1n0a.pdb

2.2 Cation-pi interactions:

```

# First, create selections for aromatic residues and cations
set aromatic_residues "resname PHE TRP TYR"
set cations "resname LYS ARG and name NZ CZ"

# Create a new representation for the protein
mol delrep 0 top
mol representation NewCartoon
mol selection "protein"
mol color Structure
mol addrep top

```

```

# Add representation for aromatic residues
mol representation Licorice
mol selection $aromatic_residues
mol color Name
mol addrep top

# Add representation for cations
mol representation VDW
mol selection $cations
mol color Name
mol addrep top

# Function to calculate center of aromatic ring
proc ring_center {resid resname} {
    set center {0 0 0}
    set n 0

    switch $resname {
        PHE {
            set atoms "CG CD1 CD2 CE1 CE2 CZ"
        }
        TYR {
            set atoms "CG CD1 CD2 CE1 CE2 CZ"
        }
        TRP {
            set atoms "CD2 CE2 CE3 CZ2 CZ3 CH2"
        }
        default {
            return $center
        }
    }

    set sel [atomselect top "resid $resid and resname $resname and name $atoms"]
    set coords [$sel get {x y z}]
    $sel delete

    foreach coord $coords {
        set x [lindex $coord 0]
        set y [lindex $coord 1]
        set z [lindex $coord 2]
        set center [vecadd $center [list $x $y $z]]
        incr n
    }

    return [vecsclae $center [expr 1.0/$n]]
}

# Find and draw cation-pi interactions
set aromatic_sel [atomselect top $aromatic_residues]
set cation_sel [atomselect top $cations]

# Create graphics
graphics top delete all
graphics top color yellow

foreach aro_resid [$aromatic_sel get resid] aro_resname [$aromatic_sel get
resname] {
    set ring_pos [ring_center $aro_resid $aro_resname]

    foreach cat_resid [$cation_sel get resid] cat_pos [$cation_sel get {x y z}]
{

```

```

    set dist [veclength [vecsub $cat_pos $ring_pos]]

    # Check if distance is within cutoff (6 Angstroms)
    if {$dist < 6.0} {
        graphics top cylinder $ring_pos $cat_pos radius 0.3 resolution 20
        graphics top sphere $ring_pos radius 0.5 resolution 20
    }
}

$aromatic_sel delete
$cation_sel delete

# Update display
display update

```

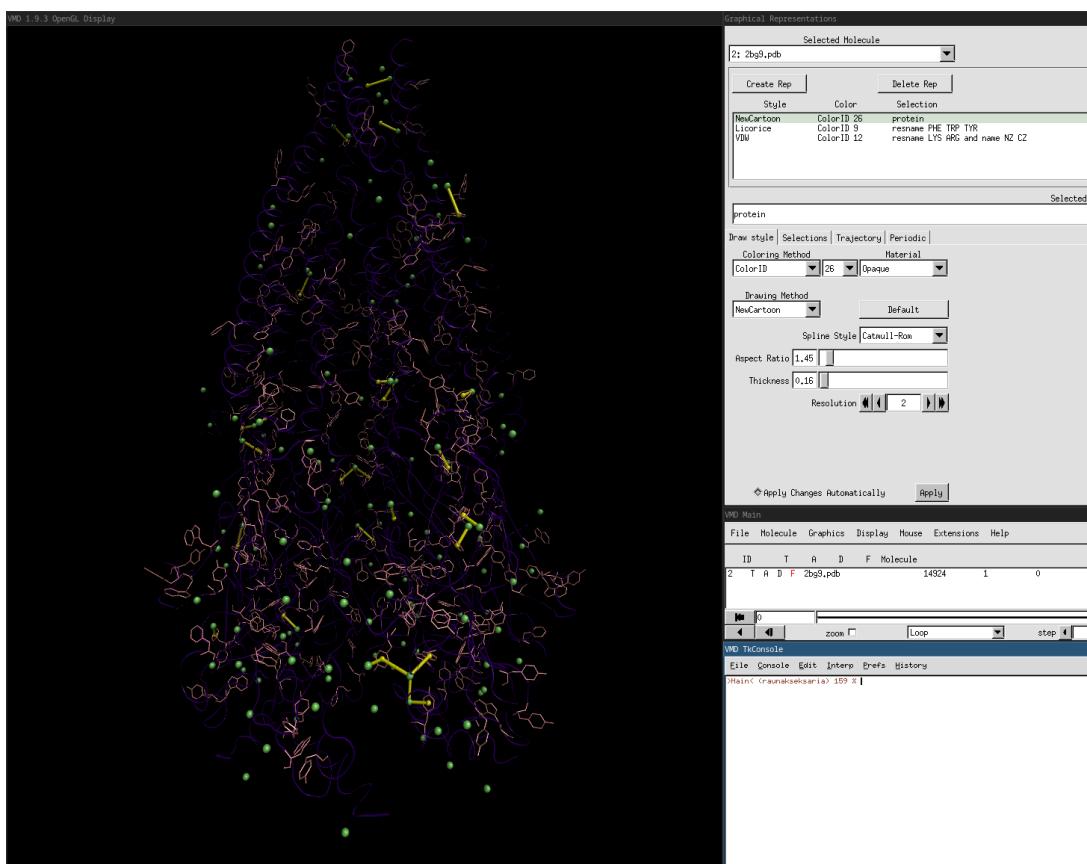


Figure 16: Cation- π interactions in 2bg9.pdb with cations and aromatic residues shown

2.3 Pi-stacking

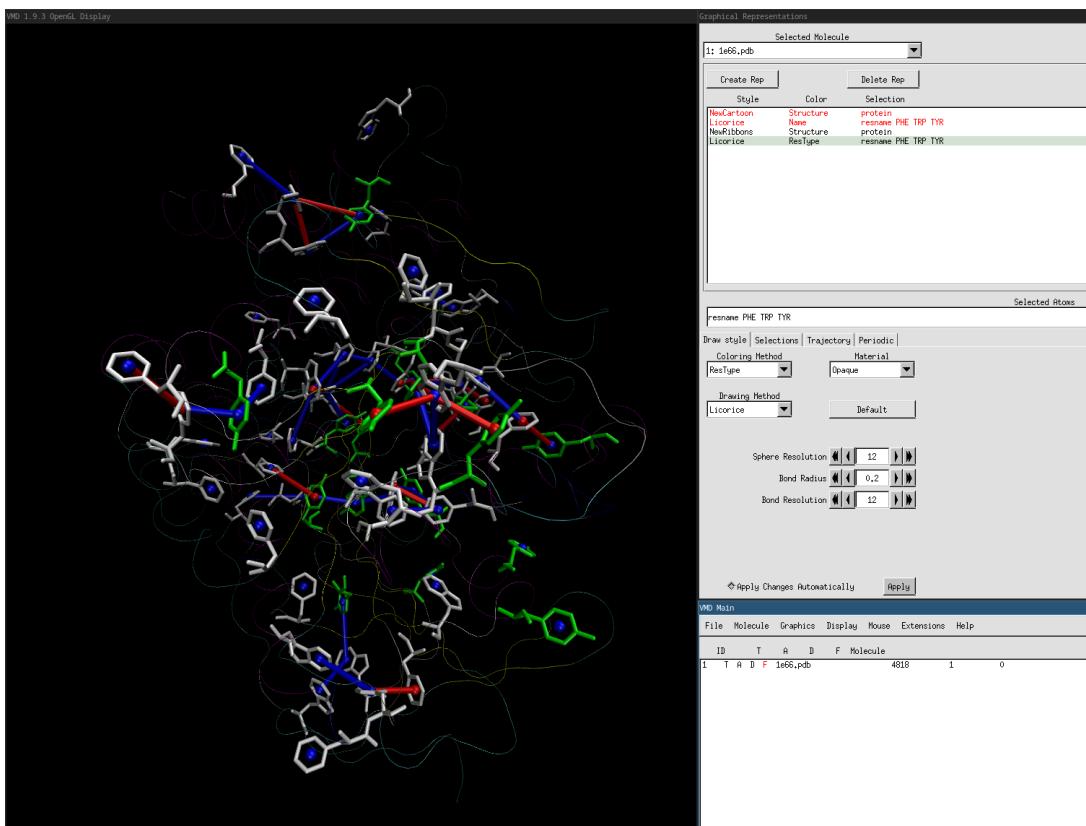


Figure 17: π - π interactions in 1e66.pdb: red is T-shaped interaction, blue is parallel interaction

2.4 Hydrophobic interactions

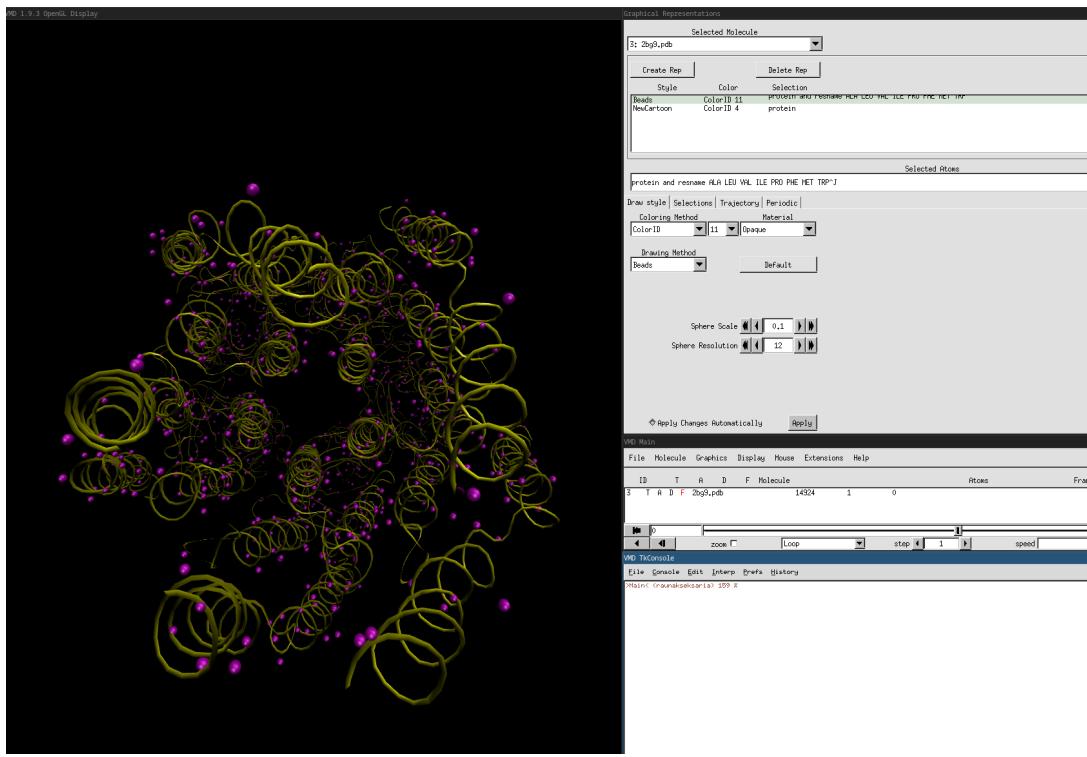


Figure 18: Hydrophobic residues shown as purple dots, and we notice that these are primarily located in the interior of the helices in 2bg9.pdb

2.5 Van-der-waals interactions

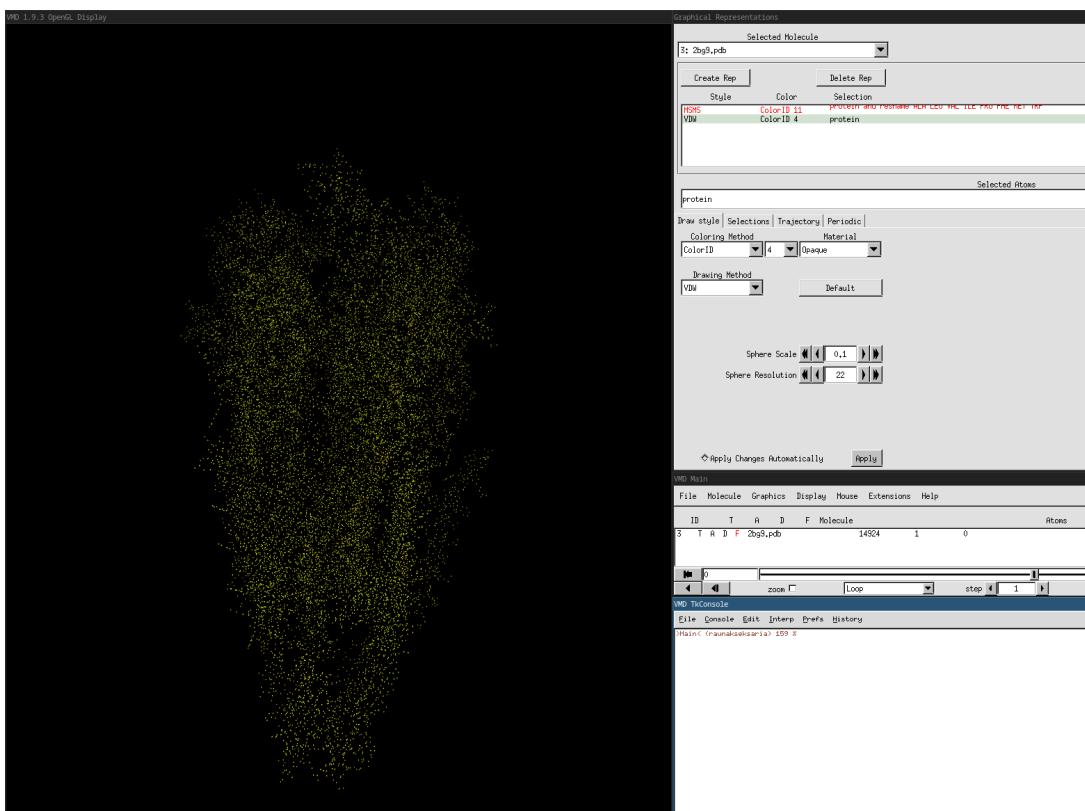


Figure 19: Van-der waal interactions in 2bg9.pdb

3 Question 3

3.1 Alpha protein

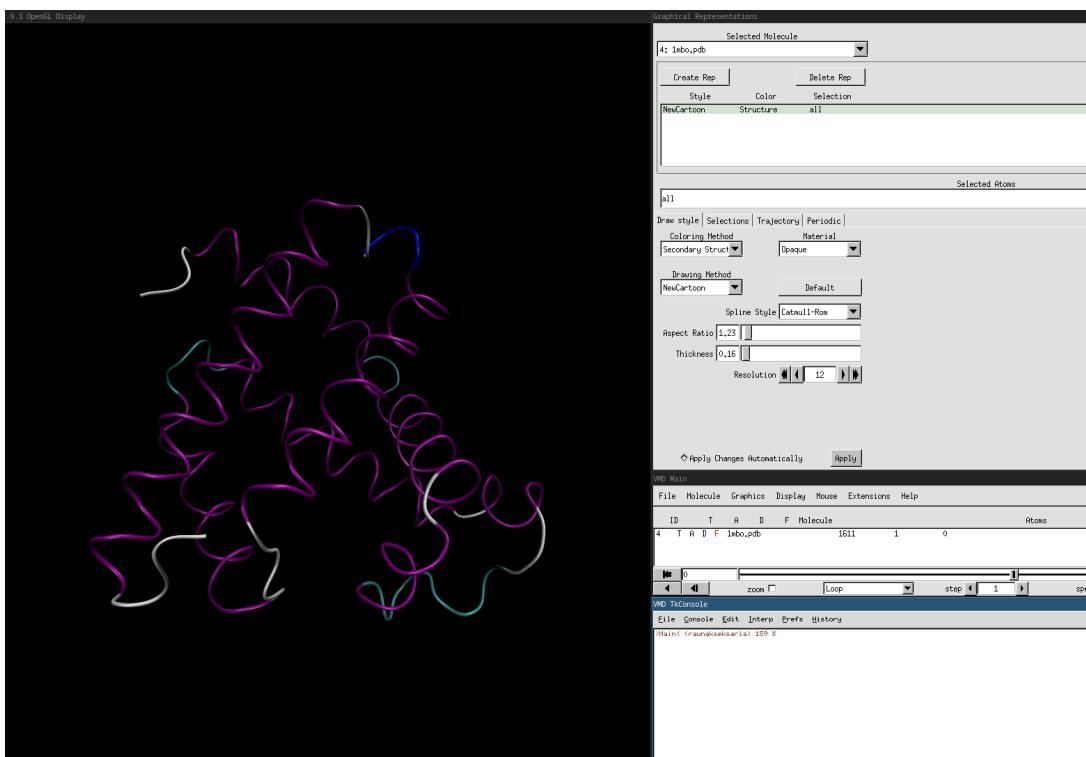


Figure 20: 1mbo mainly composed of alpha helices(shown with pink)

3.2 Beta Protein

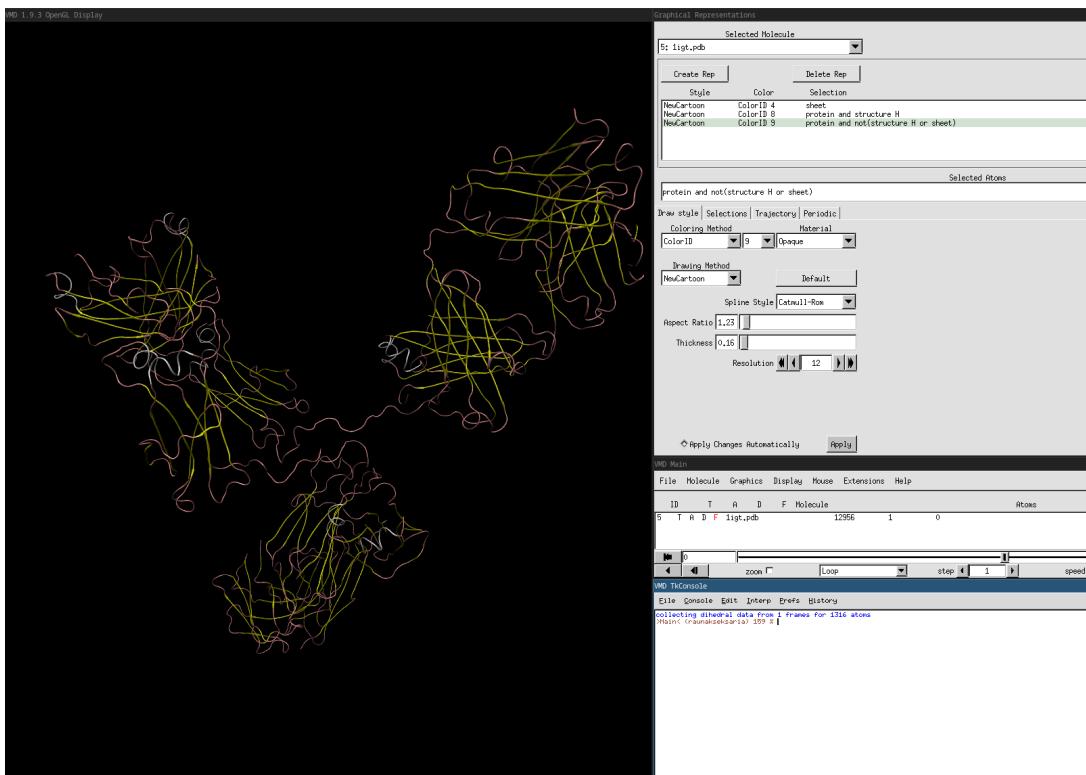


Figure 21: 1igt.pdb mainly composed of beta sheets(shown with yellow) and very little alpha helices(shown with white)

3.3 Alpha/Beta Protein

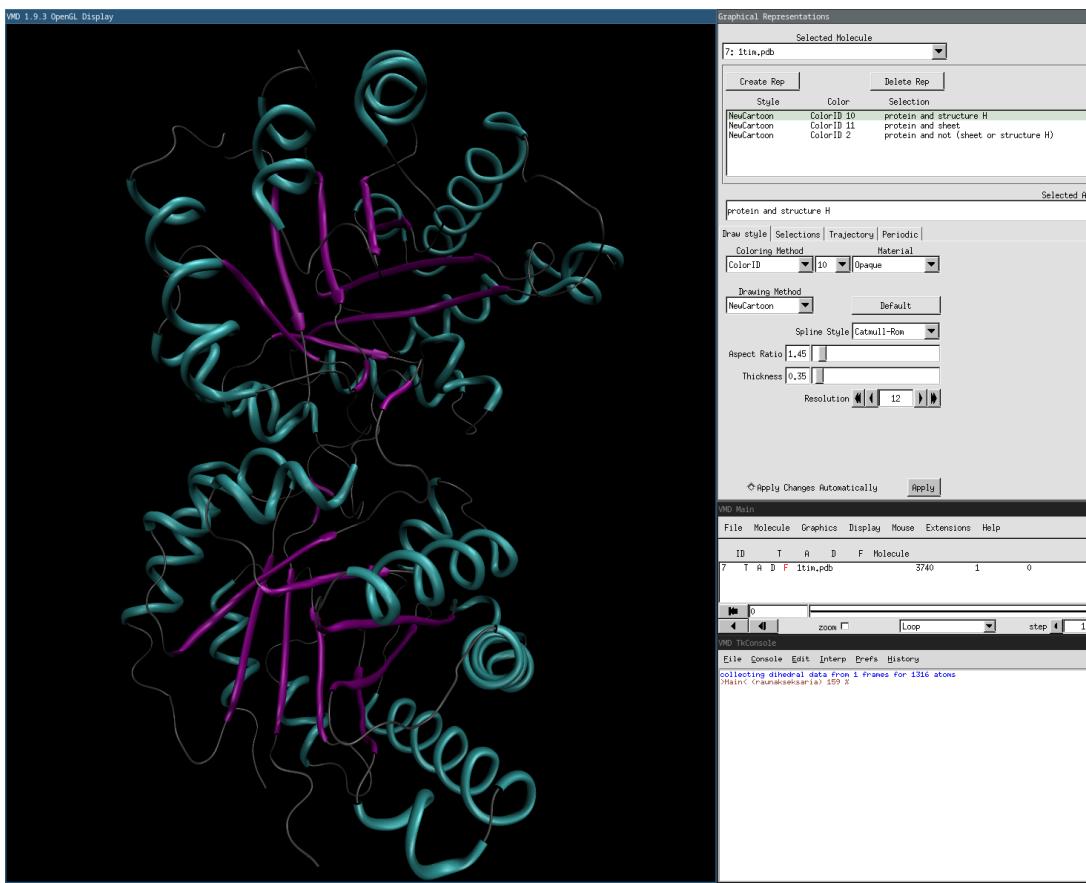


Figure 22: 1tim.pdb composed of alternating sheets(shown with pink) and alpha sheets(cyan)

3.4 Alpha + Beta Proteins



Figure 23: 7rsa.pdb composed of alpha sheets followed by beta sheets

4 Question 4:

Firstly the following code was run for getting the actual sequence:

```
import numpy as np
amino_acids = np.array(list("ACDEFGHIKLMNPQRSTVWY")) # single letter
representations
random_sequence = "".join(np.random.choice(amino_acids, 50))
print("Random Amino Acid Sequence:", random_sequence)
```

Output was:

```
SIAHETCTICCYTHRNHLCFDSNEGDKTIINAHEVKPRASQDFGWLGA
```

For the next part, I used AlphaFoldServer

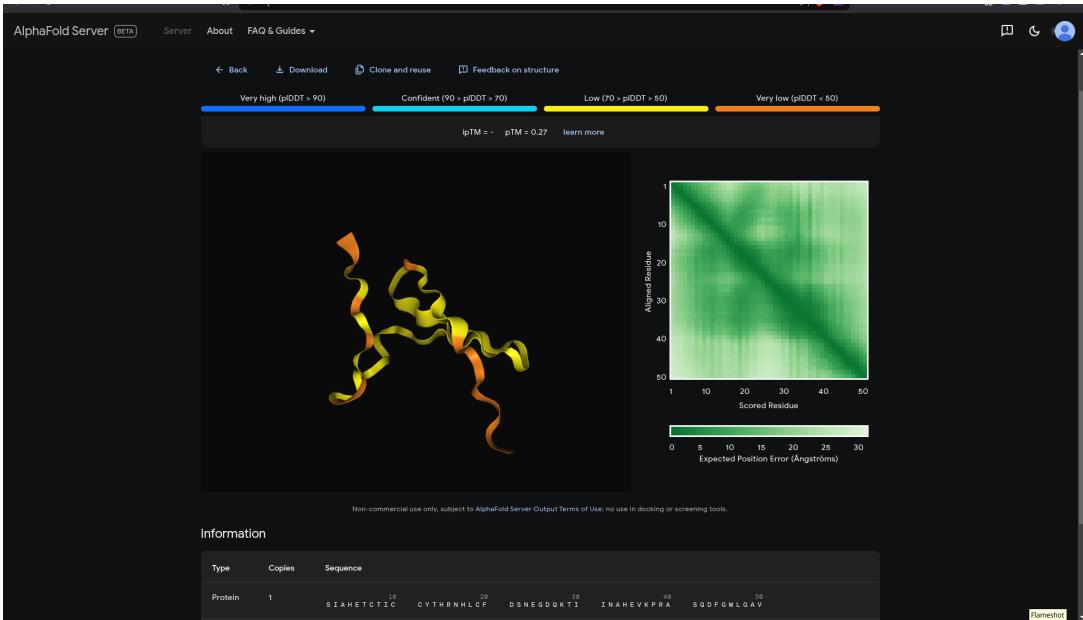


Figure 24: Results

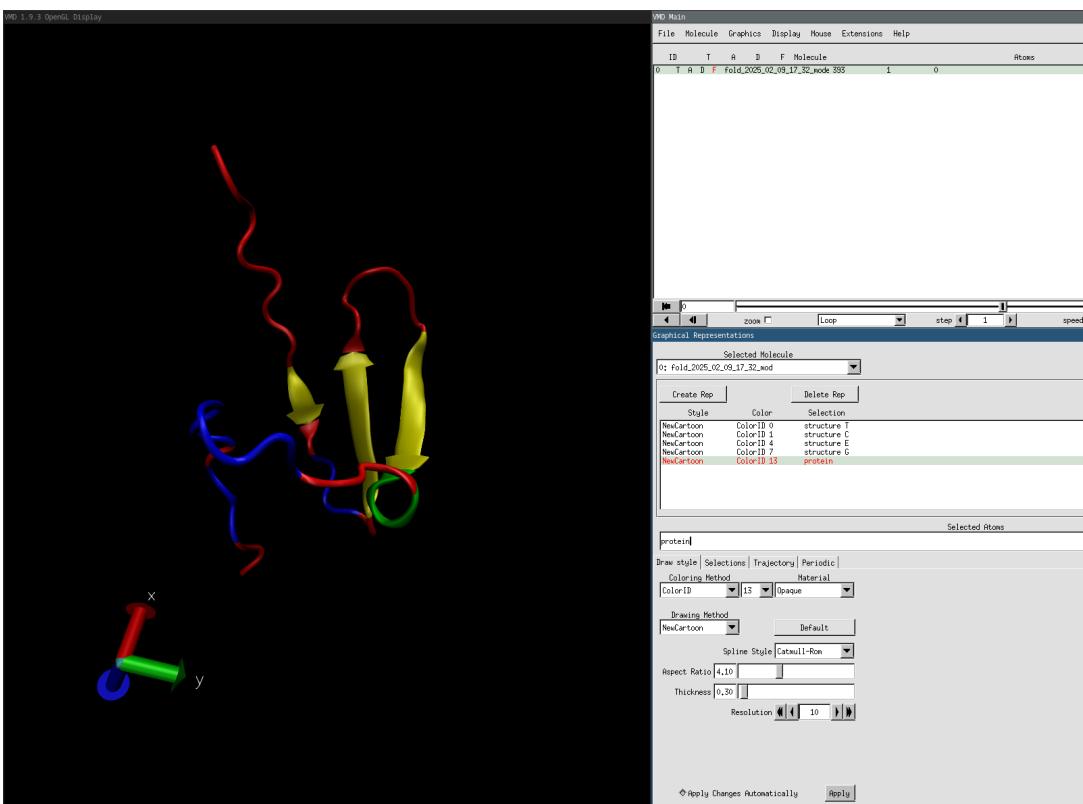


Figure 25: Downloaded the file for better understanding structure(color coded): model 0

The structure is random because:

Natural proteins have evolved to fold into stable, functional structures, while random sequences lack these evolutionary constraints. Without selective pressure, there is no guarantee that the sequence will adopt a well-defined tertiary structure.

Proteins rely on a balance of hydrophobic and hydrophilic residues to form stable cores and interfaces. A random sequence may have an irregular distribution of these residues, preventing the formation of stable secondary structures like α -helices and β -sheets.

Many random sequences resemble intrinsically disordered proteins (IDPs), which lack a stable folded structure. If the sequence does not have enough hydrophobic interactions or stabilizing hydrogen bonds, it remains in a dynamic, flexible state.

In functional proteins, loops and turns connect structured regions in a precise way. A random sequence may contain many loops and turns without stabilizing interactions, leading to a disordered, unstructured conformation.

Charged residues (e.g., D, E, K, R) may be unfavorably positioned, causing electrostatic repulsion. Bulky residues (e.g., W, F, Y) might create steric hindrance, preventing tight packing.

5 Question 5:

Firstly, to find the globular protein, SCOP2 was used, and Histone fold(2000114) was selected. From there, 1A7W was chosen.

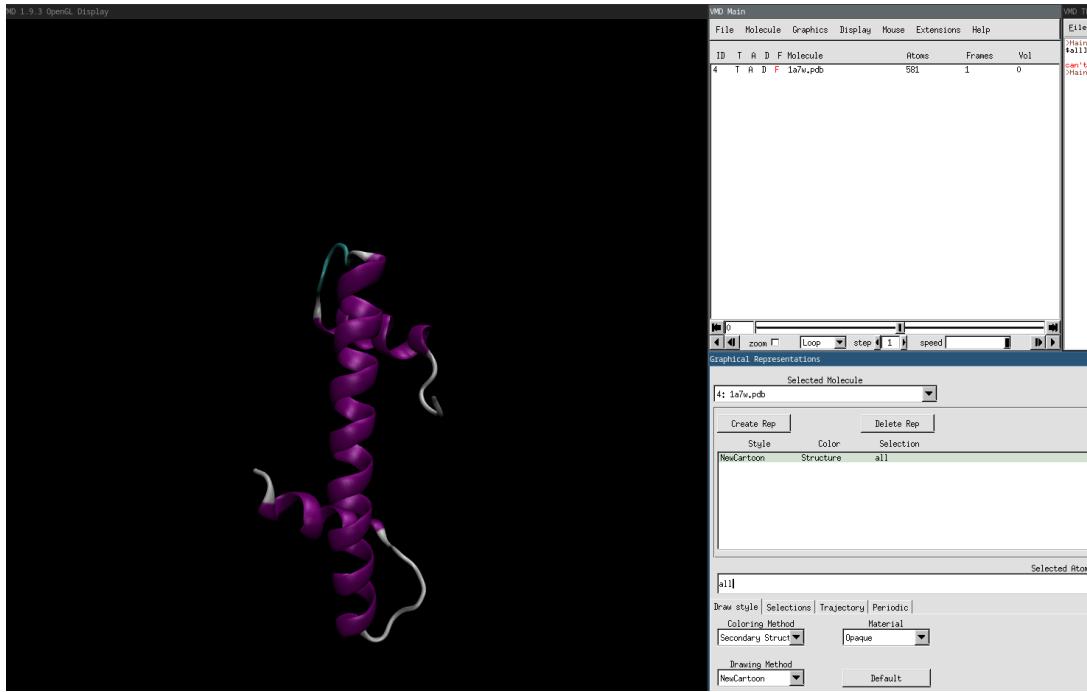


Figure 26: Basic visualisation of 1A7W.pdb

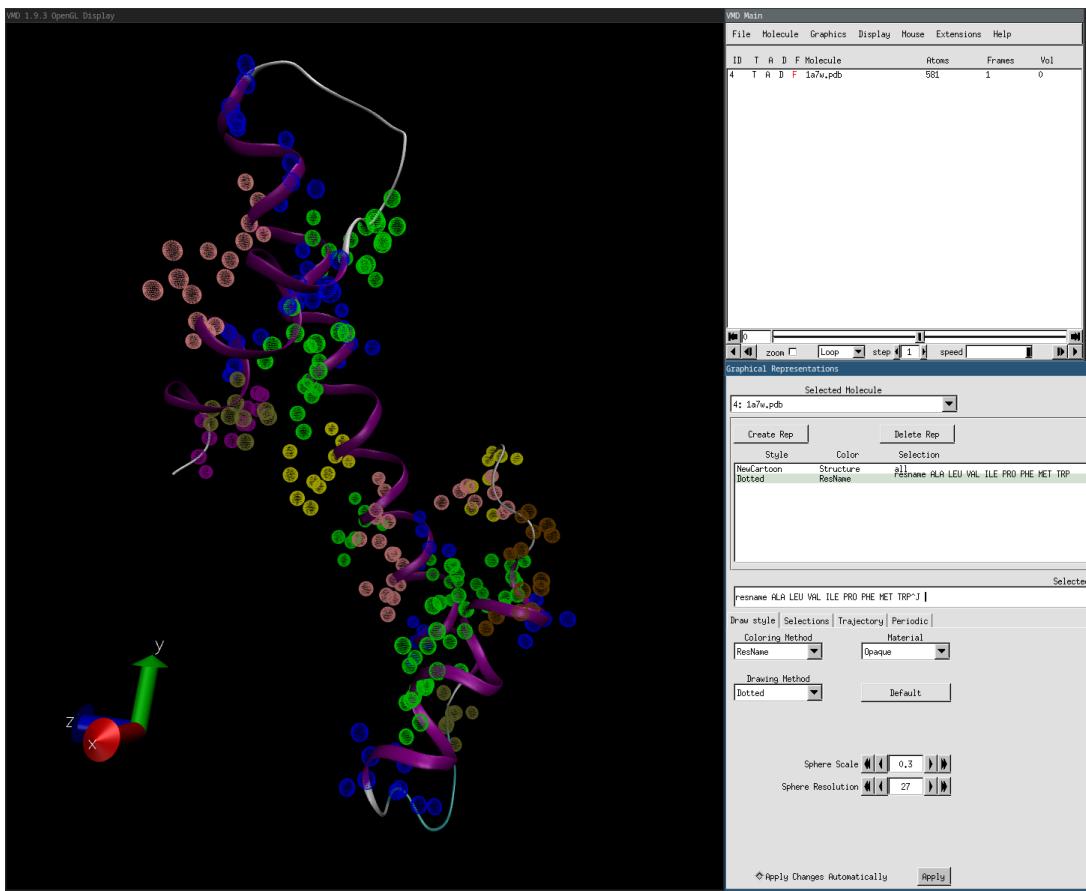


Figure 27: Hydrophobic core(using Dotted representation)

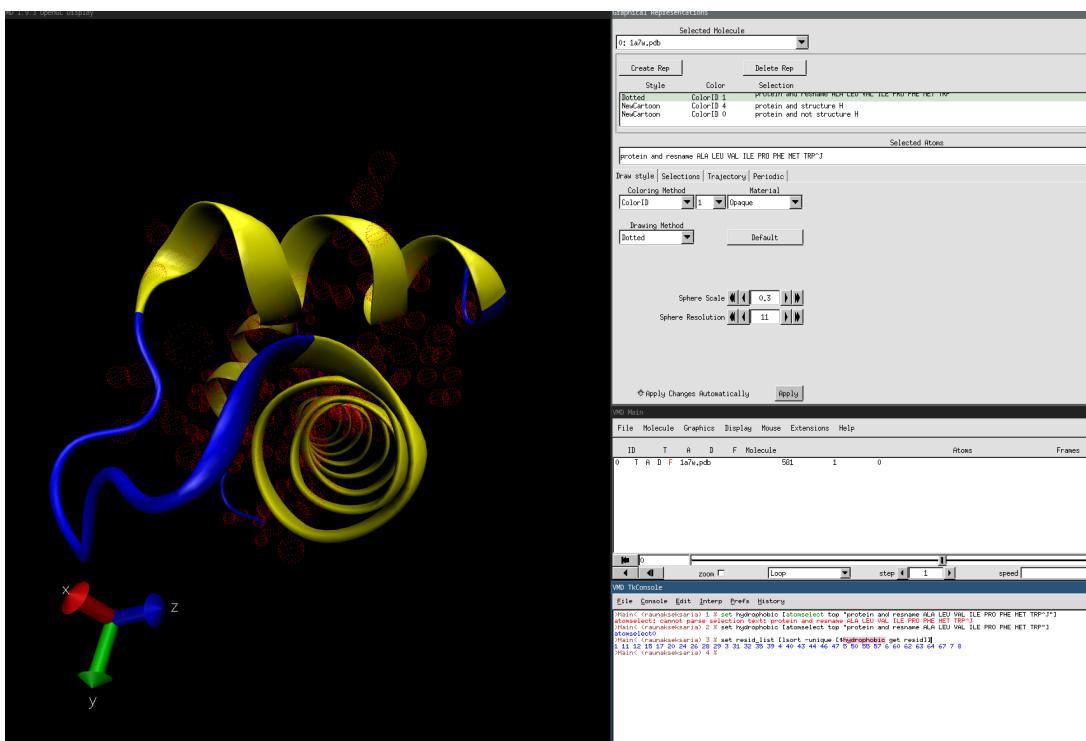


Figure 28: Hydrophobic core(using Dotted representation 2)

```
set hydrophobic [atomselect top "protein and resname ALA LEU VAL ILE PRO PHE  
MET TRP"]  
set resid_list [$hydrophobic get resid]
```

Output:

```
1 11 12 15 17 20 24 26 28 29 3 31 32 35 39 4 40 43 44 46 47 5 50 55 57 6 60 62  
63 64 67 7 8
```

A significant amount of the molecule is made up of hydrophobic residues. The number of hydrophobic residues on the outside and inside of the helix seem almost equal. But the majority of them fall on the alpha helix.