

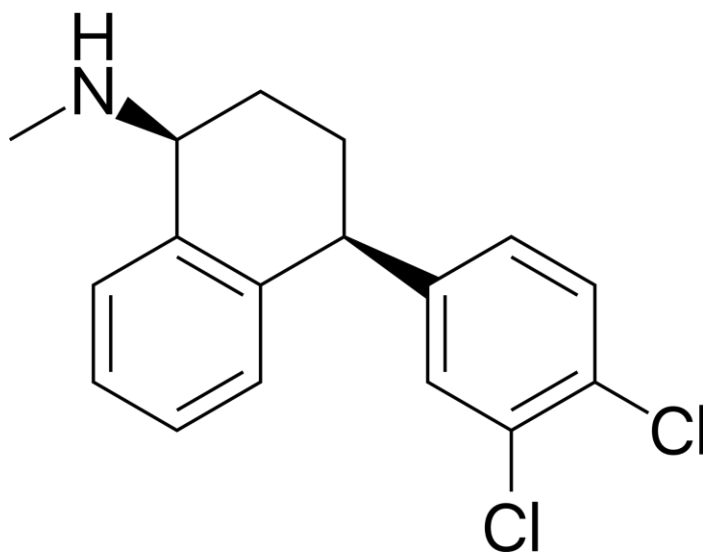
به نام خدا

تکالیف بیوانفورماتیک ساختاری

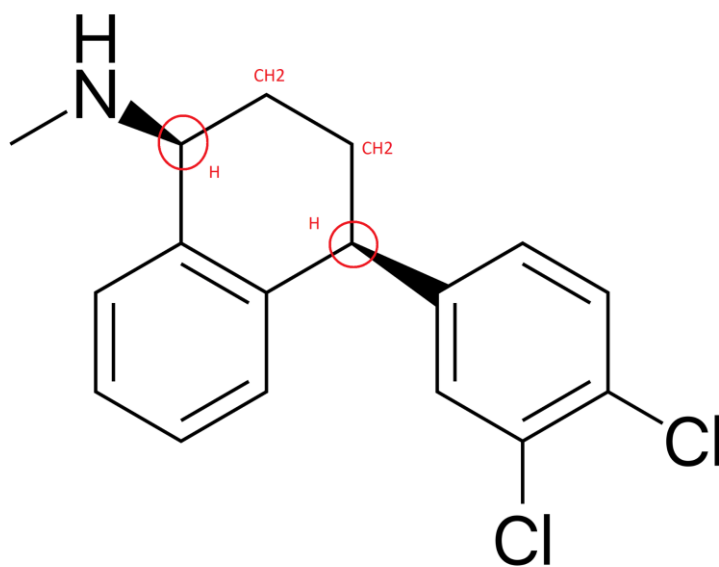
مهدی کافی ۹۹۲۱۰۷۵۳

**تکلیف اول؛** یک دارو با مرکز (یا مراکز) کایرال پیدا کرده و مرکز (یا مراکز) کایرال را در ساختار دارو مشخص کنید.

داروی انتخاب شده، **Sertraline** و یک داروی ضد افسردگی است. متأسفانه این دارو از پرمصرف‌ترین داروها در جهان است. شمای **perspective** ساختار شیمیایی این دارو در شکل زیر آورده شده است.



این ساختار دو کربن کایرال دارد که در تصویر زیر مشخص شده‌اند.

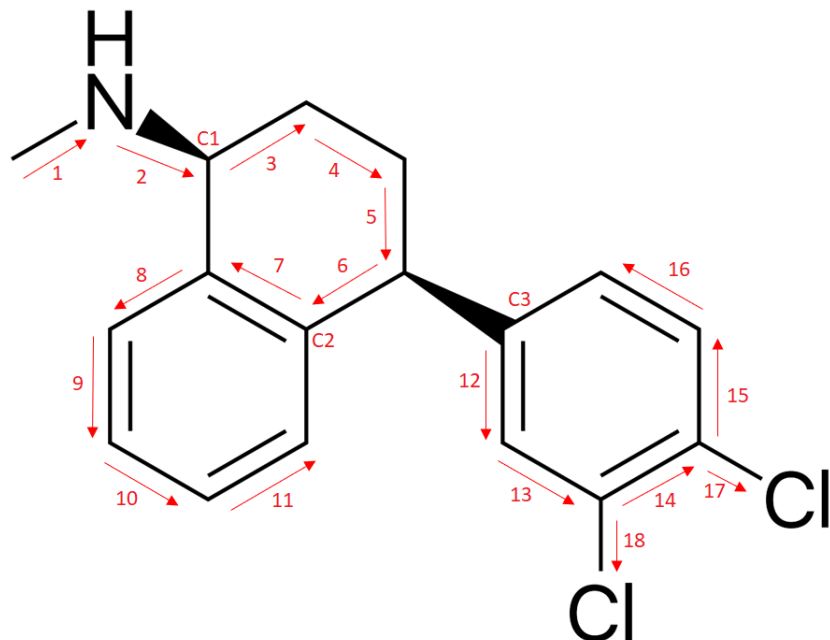


---

تکلیف دوم؛ داروی انتخابی را به صورت SMILES نشان دهید.

اگر به صورتی که مشخص شده است بر روی نمایش perspective این دارو حرکت کنیم، ساختاری SMILES به صورت زیر خواهد بود.

CN[C@H]1CC[C@H](C2=CC=CC=C12)C3=CC(=C(C=C3)Cl)Cl



---

تکلیف سوم؛ یک فایل ساختار سه بعدی پروتئین تعیین شده به وسیله NMR از پایگاه داده PDB دانلود نمایید (کد PDB پروتئین انتخاب شده را ذکر نمایید) و موارد زیر را توسط نرم افزار VMD و Tcl scripting محاسبه نمایید.

در ابتدا از طریق advanced search پایگاه داده PDB استفاده می کنیم و پروتئین هایی که به وسیله Solution NMR یا Solid-state NMR تولید شده اند را جستجو می کنیم.

RCSB PDB Deposit Search Visualize Analyze Download Learn More Documentation Careers MyPDB

Search History Browse Annotations MyPDB

Use the **Advanced Search Query Builder** tool to create composite boolean queries. See the [Help](#) page for more detailed information.

Advanced Search Query Builder [Help](#)

Full Text

Structure Attribute

Experimental Method	x	equals	SOLUTION NMR	+ NOT	Count	x
OR						
Experimental Method	x	equals	SOLID-STATE NMR	+ NOT	Count	x
AND / OR Add Attribute Add Subquery				Remove Subquery		
Add Subquery						

سپس پروتئین 2M3B را انتخاب می‌کنیم.

RCSB PDB Deposit Search Visualize Analyze Download Learn More Documentation Careers MyPDB

Structure Summary 3D View Annotations Experiment Sequence Genome Versions

← NMR Ensemble →

2M3B

Serine 16 phosphorylated phospholamban pentamer, Hybrid solution and solid-state NMR structural ensemble

DOI: 10.2210/pdb2M3B/pdb BMRB: 18952

Classification: **MEMBRANE PROTEIN**

Organism(s): *Oryctolagus cuniculus*

Expression System: *Escherichia coli* BL21(DE3)

Mutation(s): No

Membrane Protein: Yes [CPH](#) [PDBTM](#) [MemProtMD](#) [mpstruc](#)

Deposited: 2013-01-15 Released: 2013-10-30

Deposition Author(s): Vostrikov, V.V., Verardi, R., Veglia, G.

Experimental Data Snapshot

Method: SOLUTION NMR

Conformers Calculated: 200

Conformers Submitted: 20

Selection Criteria: target function

Method: SOLID-STATE NMR

Conformers Calculated: 200

Conformers Submitted: 20

Selection Criteria: target function

wwPDB Validation [3D Report](#) [Full Report](#)

Metric	Percentile Ranks	Value
Clashscore		8
Ramachandran outliers		1.0%
Sidechain outliers		9.3%

Macromolecule Content

- Total Structure Weight: 30.90 kDa
- Atom Count: 2135
- Modelled Residue Count: 260

سپس فایل PDB این پروتئین را دانلود می‌کنیم و با نرم افزار VMD این پروتئین را load می‌کنیم.

VMD TkConsole

File Console Edit Interp Prefs History Help

Main console display active (Tcl8.5.6 / Tk8.5.6)

```
(VMD) 1 % pwd
C:/Program Files (x86)/University of Illinois/VMD
>Main< (VMD) 2 % cd {W:\code\StructuralBio\Homework\HW3}
>Main< (HW3) 3 % mol load pdb 2m3b.pdb
0
>Main< (HW3) 4 %
```

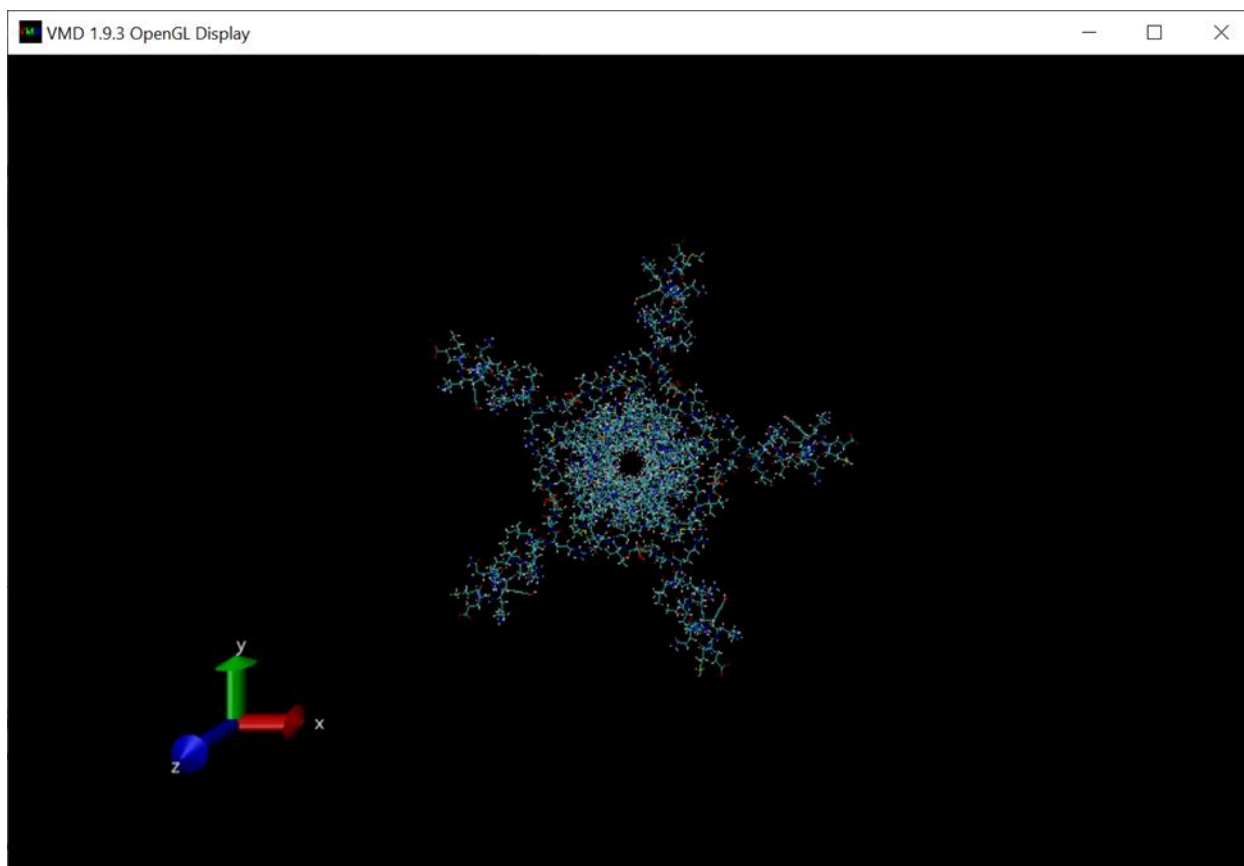
VMD Main

File Molecule Graphics Display Mouse Extensions Help

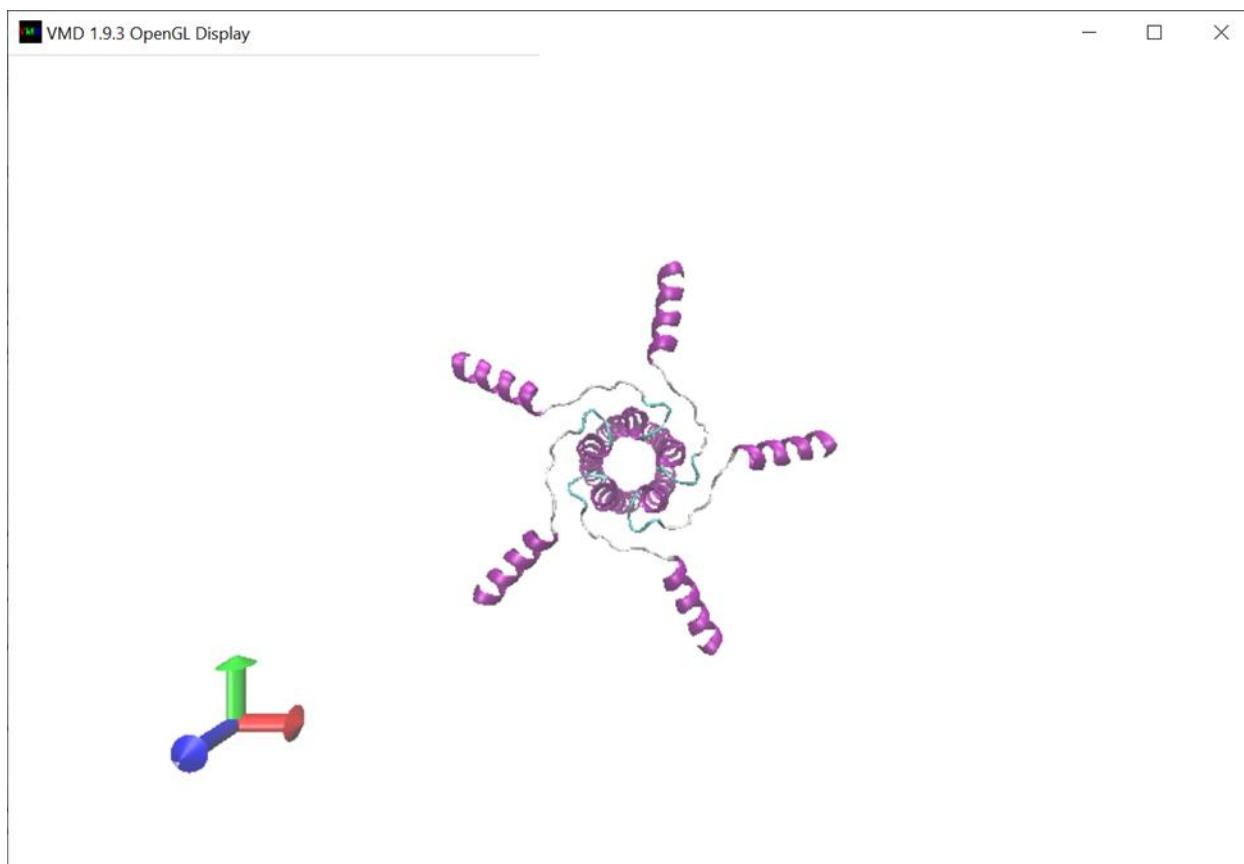
ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	2m3b.pdb	4470	20	0

19

zoom ☐ Loop  step  speed



سپس با تغییر دادن تنظیمات نمایش ساختار، در ابتدا رنگ پس‌زمینه را سفید می‌کنیم و سپس نوع نمایش را NewCartoon و نحوه رنگ آمیزی را بر اساس ساختار دوم قرار می‌دهیم. در نتیجه نمایش پروتئین به صورت زیر می‌شود.



- توالی اسیدهای آمینه و تعداد هر نوع اسید آمینه (۲۰ نوع اسید آمینه) در آن پروتئین محاسبه گردد.

برای نمایش توالی اسیدهای آمینه کافی است که مقادیر `atomselect` را در متغیر `1_sel` ذخیره کنیم و سپس از این متغیر مقادیر `rename` را نمایش دهیم.



سپس مقادیر یکتا از این لیست را در لیست unique\_residues میریزیم.

```
VMD TkConsole
File Console Edit Interp Prefs History Help
>Main< (HW3) 27 % lsort -unique $res_list
ALA ARG ASN CYS GLN GLU ILE LEU LYS MET PHE PRO SEP SER THR TYR VAL
>Main< (HW3) 28 % set unique_residues [lsort -unique $res_list]
ALA ARG ASN CYS GLN GLU ILE LEU LYS MET PHE PRO SEP SER THR TYR VAL
unique_residues unknown_handler_order unknown_handlers
unique_residues unknown_handler_order unknown_handlers
>Main< (HW3) 29 % puts $unique_residues
ALA ARG ASN CYS GLN GLU ILE LEU LYS MET PHE PRO SEP SER THR TYR VAL
>Main< (HW3) 30 % |
```

سپس از روی این لیست یک دیکشنری که ساختاری به صورت key-value میسازیم که keyها در این دیکشنری اسامی اسیدهای آمینه و valueها تعداد هر اسید آمینه است که در ابتدا همگی صفر هستند.

```
VMD TkConsole
File Console Edit Interp Prefs History Help
>Main< (HW3) 57 % puts $unique_residues
ALA ARG ASN CYS GLN GLU ILE LEU LYS MET PHE PRO SEP SER THR TYR VAL
>Main< (HW3) 58 % foreach res $unique_residues {dict set res_count $res 0}
>Main< (HW3) 59 % puts $res_count
ALA 0 ARG 0 ASN 0 CYS 0 GLN 0 GLU 0 ILE 0 LEU 0 LYS 0 MET 0 PHE 0 PRO 0 SEP 0 SE
R 0 THR 0 TYR 0 VAL 0
>Main< (HW3) 60 %
```

سپس با توجه به اینکه این پروتئین تنها ۱۷ اسید آمینه از ۲۰ اسید آمینه را دارد، ۳ اسید آمینه دیگر را به این دیکشنری اضافه می‌کنیم و تعداد اعضای این دیکشنری به ۲۰ میرسد. سپس بر روی اسامی اسیدهای آمینه پروتئین حرکت می‌کنیم و هر اسید آمینه را که دیدیم یک واحد به value آن اسید آمینه در دیکشنری اضافه می‌کنیم. در این صورت در انتهای کار تعداد هر اسید آمینه در دیکشنری ذخیره شده است. همانطور که دیده می‌شود تعداد ۳ اسید آمینه‌ای که در این پروتئین وجود نداشتند در انتها نیز برابر با صفر است.







در متغیر **n\_structures** قرار می‌دهیم. سپس ساختارهای دوم یکتا را به دست آورده از روی آنها یک دیکشنری می‌سازیم و مقدار **value** تمام آنها را برابر با صفر قرار می‌دهیم. سپس لیست **sec\_struct** را پیمایش می‌کنیم و هر ساختار را که می‌بینیم مقدار **value** آن ساختار که نشان دهنده تعداد آن ساختار در پروتئین است را یک واحد زیاد می‌کنیم. در نهایت کافیت که مقادیر محاسبه شده را بر تعداد کل ساختارها تقسیم و در عدد ۱۰۰ ضرب کنیم تا درصد هر ساختار دوم در این پروتئین محاسبه شود.

```

q3_procedure.tcl X
W: > code > StructuralBio > Homework > HW3 > q3_procedure.tcl
1  proc sec_struct_proportion {pdbfile frame_number} {
2      set mol_id [mol load pdb $pdbfile]
3      set sel_3 [atomselect $mol_id "all" frame $frame_number]
4      set sec_struct [$sel_3 get {structure}]
5      set n_structures [llength $sec_struct]
6      set unique_structs [lsort -unique $sec_struct]
7      foreach struct $unique_structs {dict set struct_proportion $struct 0.0}
8      foreach struct $sec_struct {dict set struct_proportion $struct [expr 1+[dict get $struct_proportion $struct]]}
9      foreach key [dict keys $struct_proportion] {dict set struct_proportion $key [expr [dict get $struct_proportion $key]/$n_structures*100]}
10     puts $struct_proportion
11 }
12

```

نتیجه این **procedure** به صورت زیر است. به طور مثال ۴۲/۸۰ درصد از ساختارهای دوم این پروتئین را **helix**ها تشکیل می‌دهند.

```

74 VMD TkConsole
File Console Edit Interp Prefs History Help
>Main< (HW3) 263 % source q3_procedure.tcl
>Main< (HW3) 264 % sec_struct_proportion 2m3b.pdb 1
C 11.968680089485458 H 80.42505592841164 T 7.606263982102908
>Main< (HW3) 265 % |

```

- زوایای **phi** و **psi** برای همه مدل (فریم) های پروتئین محاسبه گردد و نمودار را ماچاندران رسم شود.

در ابتدا محتوای **atomselect** این مولکول را در متغیر **4\_sel** می‌ریزیم. سپس مقادیر زوایای **phi** و **psi** را با دستور **get** استخراج می‌کنیم و نمایش می‌دهیم.



سپس این زوایا را در لیست phi\_psi ذخیر کردیم و آنها را در فایل phi\_psi نوشتیم. این فایل به صورت زیر است.

[illegible]

با قطعه کد زیر که به زبان پایتون نوشته شده است، این زوایا را رسم کردیم.



```

import re
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme()

with open("phi_psi.txt") as file:
    content = file.read()

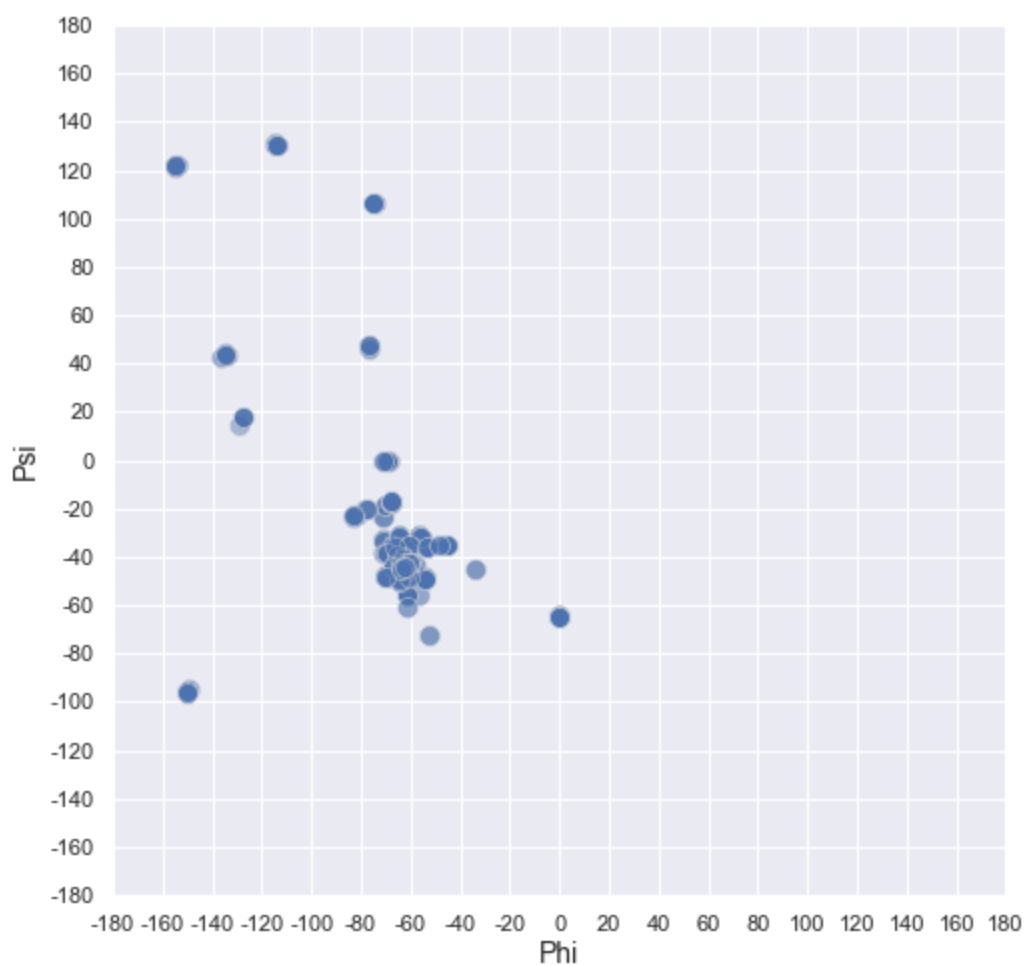
pattern = "{(.+?) (.+?)}"

points = re.findall(pattern=pattern, string=content)
points = [(float(phi), float(psi)) for (phi, psi) in points]
phis, psis = [], []
for (phi, psi) in points:
    phis.append(phi)
    psis.append(psi)

sns.set(rc={'figure.figsize':(8, 8)})
ax = sns.scatterplot(x=phis, y=psis, s=100, alpha=0.05)
ax.set_xticks(range(-180, 181, 20))
ax.set_xticklabels(list(range(-180, 181, 20)))
ax.set_yticks(range(-180, 181, 20))
ax.set_yticklabels(list(range(-180, 181, 20)))
ax.set_xlabel("Phi", fontsize=14)
ax.set_ylabel("Psi", fontsize=14)
plt.savefig("phi_psi.png")

```

نمودار تولید شده توسط این قطعه کد به صورت زیر است.



---

- یک `procedure` جدید طراحی گردد که فایل `pdb` را `load` کرده و همه مدل ها در فایل `PDB` را به فریم `align 0` نماید و مقدار `RMSD` را برای آن ها محاسبه نماید.

در ابتدا پروتئین را با استفاده از آدرس فایل آن که ورودی `procedure` است `load` می کنیم و شماره `ID` این مولکول را در متغیر `mol_id` می ریزیم. سپس محتوای `atomselect` این پروتئین را در متغیر `5_sel` می ریزیم. سپس با `get` کردن از اطلاعات `molinfo` تعداد فریم های `NMR` این پروتئین را در متغیر `n_frames` می ریزیم. سپس در یک حلقه بر روی فریم ها می چرخیم و با داشتن هر فریم اطلاعات آنرا در متغیر `cur_frame` می ریزیم. این فریم را بر فریم صفر `fit` می کنیم و ماتریس تبدیل را محاسبه می کنیم. سپس با داشتن این ماتریس فریم جدید را بر فریم صفر `align` می کنیم و سپس مقدار `RMSD` بین فریم `align` شده و فریم صفر را محاسبه و چاپ می کنیم.

```
phi_psi_plot.ipynb  q5_align_rmsd.tcl X
q5_align_rmsd.tcl
1  proc align_rmsd {pdbfile} {
2      set mol_id [mol load pdb $pdbfile]
3      set sel_5 [atomselect $mol_id "all" frame 0]
4      set n_frames [molinfo top get numframes]
5
6      for {set frame_idx 0} {$frame_idx < $n_frames} {incr frame_idx} {
7          set cur_frame [atomselect $mol_id "all" frame $frame_idx]
8          set transform_matrix [measure fit $cur_frame $sel_5]
9          $cur_frame move $transform_matrix
10         set rmsd [measure rmsd $cur_frame $sel_5]
11         puts "Frame: $frame_idx | rmsd: $rmsd"
12     }
13 }
14
```

نتیجه اجرای این procedure برای پروتئین M3B2 به صورت زیر است.

```
VMD TkConsole
File Console Edit Interp Prefs History Help
>Main< (HW3) 20 % source q5_align_rmsd.tcl
>Main< (HW3) 21 % align_rmsd 2m3b.pdb
Frame: 0 | rmsd: 0.0
Frame: 1 | rmsd: 7.4209489822387695
Frame: 2 | rmsd: 5.448541164398193
Frame: 3 | rmsd: 8.609509468078613
Frame: 4 | rmsd: 5.758419513702393
Frame: 5 | rmsd: 4.444986820220947
Frame: 6 | rmsd: 3.7583658695220947
Frame: 7 | rmsd: 5.82644510269165
Frame: 8 | rmsd: 7.900402069091797
Frame: 9 | rmsd: 6.323787212371826
Frame: 10 | rmsd: 9.552952766418457
Frame: 11 | rmsd: 6.280325889587402
Frame: 12 | rmsd: 7.044864177703857
Frame: 13 | rmsd: 5.437060832977295
Frame: 14 | rmsd: 5.3072509765625
Frame: 15 | rmsd: 5.558046340942383
Frame: 16 | rmsd: 4.333827018737793
Frame: 17 | rmsd: 9.192429542541504
Frame: 18 | rmsd: 5.261538028717041
Frame: 19 | rmsd: 10.813769340515137
>Main< (HW3) 22 %
```