PyMOL Tutorial — Neuraminidase Inhibitors (DANA, Zanamivir, Oseltamivir, Peramivir)

Learning objectives

1. What NA inhibitors are and why they treat influenza.
2. How the natural substrate (sialic acid) binds vs. inhibitors (transition-state analogue logic).
3. How to load PDBs, show cartoons/surfaces/ligands, find H-bonds, and superpose proteins.
4. How common resistance mutations disrupt binding.
5. How peramivir's design aims to be robust.
6. How to save your work, write a repeatable `.pml` script, and export a short movie.

Before you start

- **Install PyMOL** (open-source or incentive builds work).
- **Internet access** to fetch PDBs inside PyMOL
- To download PDB files directly and inspect entry pages got to the Protein Data Bank website.
- We'll use these PDB IDs:
    - **Sialic acid (substrate)**: `2BAT`
    - **DANA (prototype, TS analogue)**: `1F8B`
    - **Zanamivir**: `2HTQ`
    - **Oseltamivir**: `2HU4`
    - **Peramivir**: `2HTU`
    - **Resistance mutant (H274Y)**: `3CL0` (with oseltamivir)

---

1) Very short scientific background (read this first)

Neuraminidase (NA) is an influenza surface enzyme that clips off terminal **sialic acid** from host glycans. When NA works, new virions don't stick to the cell; they're released. If we **inhibit NA**, virions remain tethered, and the infection struggles to spread.

- The **natural substrate** is **sialic acid** (5-acetyl-neuraminic acid or **Neu5Ac**). During catalysis NA stabilizes a **distorted, positively polarized transition state**.
- Medicinal chemists asked: *What if we build small molecules that look like that transition state?*
    - **DANA (Neu5Ac2en)** was the first **transition-state analogue (TSA)**: it binds better than sialic acid because it resembles the catalytically preferred geometry/charge distribution.
    - **Zanamivir** improves on DANA by adding a **4-guanidino** group to form extra ionic H-bonds with acidic residues in NA.
    - **Oseltamivir** changes the "sugar" into a **cyclohexene** and adds a **hydrophobic side chain** that fills a nearby pocket — enabling **oral dosing** (fewer strong charges) while keeping potency.

- **Peramivir** keeps a **strong polar network** (like zanamivir) but also occupies pocket volume; it's given **IV** and often shows **tight/long residence**.

Resistance. NA mutates. If a mutation reshapes the hydrophobic pocket or weakens key ionic contacts, some drugs bind worse. Canonical examples include **H274Y (N1)**, **E119V**, **R292K** — each reduces binding in a slightly different way.

---

2) Meet the active site with the natural substrate (sialic acid)

Goal: Load an NA–sialic acid complex, find the ligand, and glance at the pocket.

1. **Reset and fetch the complex**

   In PyMOL's command line (bottom), paste and run:

   ```
   reinitialize
   ```

   This isn't really needed now, but it is useful to know - it will reset PyMOL to a clean, initial state, deleting all loaded objects.

   Now a few configuration commands for nicer viewing:

   ```
   set stick_radius, 0.15
   set valence, 0
   set stick_h_scale, 1.0
   ```

   Let's get our first PDB file in PyMOL, using the `fetch` command:

   ```
   fetch 2bat, async=0
   ```

   this commands retrieves (fetches) the PDB file with code `2bat` from the PDB, and displays in on the PyMOL window. You will notices that there portions of the object on screen represented in different ways: a `cartoon` representation of the protein, sticks for the sugar chains (and the ligand), spheres for ions (in this case Ca++) and small red dots for solvent molecules (crystallographic waters).

   Let's start by removing the water molecules to improve clarity:

   ```
   remove solvent
   ```

   and change the cartoon color:

   ```
   set cartoon_color, gray80
   ```

Notice the sticks chains (sugar chains) seem to "float", because they appear disconnected from the protein - this is not the case, just a consequence using different representations for protein a sugar chains.

2. **Show the ligand and pocket residues**

   Now, let's zoom in on the sialic acid substrate:

```
zoom SIA/
```

   "SIA" is the residue name for sialic acid (you can check this on the PDB site entry for 2BAT). Now, the ligand should be filling your screen. To get a better view, use the right mouse button to zoom in or out, the left mouse button to rotate, and the mouse wheel to change the size of the vicing volume (hiding more or less of the sialic acid surroundings). Notice how the scene rotates around the "SIA" ligand.

   To analyze contacts, let's select residues within a short distance of the ligand and show them as sticks:

```
select site, byres (SIA/ around 6) and polymer
```

   The above command creates a *name selection* called `site`. Notice how this name appears on the right-hand panel of the application. And now let's show the selected nearby residues as sticks:

```
color cyan, site
```

   To visualize the shape of the active site pocket as surface, we use:

```
show surface, byres (SIA/ around 4)
set transparency, 0.5
```

   - `SIA/` catches sialic acid.
   - `byres (SIA/ around 6)` grabs entire residues within 6 Å.
   - A semi-transparent surface helps beginners “see the pocket-shaped cave”.

   Note: to avoid using the `/` at the end of `SIA`, and also to make SIA appear in the object list, we can do this:

```
select SIA, SIA/
```

This will simply create a *name selection* called `SIA` and thus add it to the object menu on the right-hand side. All the above commands that used `SIA/`could now be run with just `SIA`. Important to note that you should only do this with names that are *unique*, like ligands.

3. **Look at polar contacts (approximate way to view H-bonds/ion pairs)**

   H-bonds in PyMOL can be visualized in various ways. A simple approach is to use the `distance` (which draws dashed lines joining atoms), select the donor and acceptor polar atom types and with a proper cut-off distance (to prevent lines to be drawn between atoms too far apart):

   ```
   dist hb_sia, polymer, SIA, cutoff=3.6, mode=2
   hide labels, hb_sia
   orient SIA
   ```

   - You'll see dashed lines from protein N/O atoms to ligand atoms (within a typical H-bond distance/angle).

   What to observe: a nest of **arginines and acidic residues** around the carboxylate and ring oxygens of sialic acid. The ligand often appears **slightly distorted**, hinting at the **transition state** NA stabilizes.

---

3) See a prototype TSA: DANA (why it binds better than sialic acid)

Now we are going to load another neuraminidase in complex with a *transition state analogue* (**TSA**), DANA.

Goal: Load DANA, compare its contacts with sialic acid's.

Optional: We can hide the current object by clicking on the `2BAT` bar on the right-hand side panel, or by issuing the command:

```
disable 2BAT
```

just for clarity. We can re-enable viewing at any time by clicking again on the `2BAT` bar in the object menu.

1. **Fetch and show the DANA complex**

   ```
   fetch 1f8b, async=0
   remove solvent and 1f8b
   set cartoon_color, slate, 1f8b
   select DANA, DAN/
   show sticks, DANA
   color orange, DANA
   ```

2. **Superpose DANA complex onto the sialic-acid complex**

Now we want to compare the two complexes. First, enable 2bat with:

```
enable 2bat
```

and then *superimpose* the two molecules with the command:

```
align 1f8b and name CA, 2bat and name CA
```

- `align` optimally fits atoms of the objects, minimizing their distances. The "and name CA" part forces `align` to use fit only the Cα, ignoring remain protein atoms. Notice how similar the two structures are. Now that the pockets are aligned, is much easier to compare the interaction mode of the two ligands.

3. **Check polar contacts for DANA**

```
dist hb_dana, (1f8b and polymer), DANA, mode=2, cutoff=3.6
hide labels, hb_dana
```

DANA **mimics** the **cationic/planar TS** better than Neu5Ac, so the **electrostatic and geometric complementarity** is stronger. Compared to the sialic acid substrate, the hydrogen bond pattern is very similar.

---

4) Zanamivir: the 4-guanidino upgrade (stronger anchoring)

Goal: See how adding a **guanidinium** at C4 strengthens binding.

1. **Fetch zanamivir complex and display**

```
fetch 2htq, async=0
remove solvent and 2htq
set cartoon_color, salmon, 2htq
select ZMR, ZMR/
color white, ZMR
```

2. **Superpose against the sialic-acid complex (or DANA)**

```
align 2htq and name CA, 2bat and name CA
```

3. **Measure contacts**

```
dist hb_zmr, (2htq and polymer), ZMR, mode=2, cutoff=3.6
hide labels, hb_zmr
orient ZMR
```

The **4-guanidino** group in zanamivir forms **extra salt bridges/H-bonds** with acidic residues (E119/E227). That **locks** the ligand in the **TS-like pose** more firmly than natural Neu5Ac — a reason for **higher affinity** but also **lower oral bioavailability** (it's very polar).

---

5) Oseltamivir: pocket filling and oral dosing (fewer H-bonds, more hydrophobics)

Goal: See a different strategy: **fill a hydrophobic pocket** to gain binding energy while keeping a scaffold suitable for **oral** delivery.

1. **Fetch and show the oseltamivir complex**

In this case, the PDB file contains two copies of a tetrameric neuraminidase. We need to remove all but one chain

```
fetch 2hu4, async=0
remove solvent and 2hu4
remove 2hu4 and not chain A
set cartoon_color, limon, 2hu4
select OTV, G39/ and 2hu4
color yellow, OTV
```

The command `remove 2hu4 and not chain A` works, because polypeptide chains in the PDB structure are labeled `A,B,C,D,...` and we want only one of them (you might check the `2UH4` entry in the PDB website).

2. **Superpose against zanamivir (to compare binding styles)**

```
align 2hu4 and name CA, 2htq and name CA
```

3. **Polar contacts (you'll likely see fewer than with zanamivir)**

```
dist hb_otv, (2hu4 and polymer), OTV, mode=2, cutoff=3.6
hide labels, hb_otv
```

4. **Show the hydrophobic pocket around the pentyl ether**

```

```

```
select site_otv, byres (OTV around 6) and 2hu4
show sticks, site_otv
show surface, byres (OTV around 4) and 2hu4
```

Compared to zanamivir, oseltamivir makes **fewer strong H-bonds** but **packs** a **hydrophobic side chain** into a **lipophilic pocket** near the “150-loop” region. This tradeoff — more hydrophobics, fewer charges — supports **oral bioavailability** while maintaining potency.

---

6) Resistance: watch how a single mutation spoils the fit (H274Y example)

Goal: Compare **wild-type** vs **H274Y** (N1) with oseltamivir bound. You’ll see the hydrophobic pocket **reshape** and the ligand fit worsen.

1. **Load the H274Y mutant**

```
fetch 3cl0, async=0
remove solvent, 3cl0
set cartoon_color, raspberry, 3cl0

select OTVm, 3cl0 and resn G39
show sticks, OTVm
color gold, OTVm
```

2. **Superpose mutant vs wild-type (2HU4)**

```
align 3cl0 and name CA, 2hu4 and name CA
```

3. **Highlight the mutation site**

```
# In N1 numbering you’ll see 274/275 variants; show a blob around the
site:
select mutSite, 3cl0 and byres (resi 274-276 and chain A)
show sticks, mutSite
color red, mutSite
label mutSite and name CA, "H274Y region"
```

4. **Compare contacts**

```
dist hb_wt, (2hu4 and polymer), OTV, mode=2, cutoff=3.6
dist hb_mut, (3cl0 and polymer), OTVm, mode=2, cutoff=3.6
hide labels, hb_wt hb_mut
```

H274Y nudges nearby residues (notably E276) and **collapses the hydrophobic cavity** where oseltamivir's **pentyl ether** sits. Result: poorer fit, fewer stabilizing contacts → **resistance**. Zanamivir, anchored by multiple **strong ionic H-bonds**, is **less** affected by this particular mutation (though other mutations can affect zanamivir binding).

Tip: If you don't have a ready PDB for a given mutation (e.g., E119V, R292K), PyMOL's **Mutagenesis Wizard** (Menu → **Wizard** → **Mutagenesis**) lets you introduce a mutation on the WT structure, preview rotamers, and then re-evaluate clashes/contacts. It's not a substitute for crystallography, but it's great for teaching geometry.

---

7) Practice exercise: analyze Peramivir (strong polar anchoring + pocket occupancy)

Goal: See a "hybrid" strategy: keep **zanamivir-like** polar anchoring **and** fill space like oseltamivir.

Start by searching the PDB for a neuramidinase structure in complex with Peramivir. Follow the previous procedure to analyze the binding pocket and the protein-ligand interactions.

Peramivir often retains a **dense ionic/H-bond network** (good against **H274Y**-style pocket reshapes) while also occupying neighboring subpockets. It's **IV** and can exhibit **tight/long residence**. Some mutations (e.g., **R292K**) still reduce binding — no single design beats all mutations — but Peramivir shows **robustness** in scenarios where Oseltamivir fails.

---

8) What those PyMOL commands actually did (gentle reference)

- `fetch <PDB>`: download a structure by PDB ID.
- `remove solvent`: hide waters/ions (you can keep them if you want to discuss bridging H-bonds).
- `show cartoon|sticks|surface`: three common representations (secondary structure, bonds, surface).
- `select <name>, <query>`: make a named selection (ligands via `resn`, pocket via `around` distance).
- `dist`: draw dashed lines between atoms (we used it to approximate H-bonds with distance/angle filters).
- `align`: superpose structures (usually on the Cα atoms).
- `orient`: center/zoom on the selection nicely.
- `color`: color an object or selection (e.g., `color cyan, site`).
- `label`: add on-screen text (atomwise; use sparingly).

Tip: Use the **"A/S/H/L"** buttons in the right panel (Action/Show/Hide/Label) when you forget a command — they show you the equivalent PyMOL operations in menus.

---

9) Scenes and movies

Scenes save camera, visibility, and coloring, perfect for presentations and teaching.

```
# Set up a nice view on zanamivir (after all your styling)
orient ZMR
scene F1, store  # save current view/visibility as scene "01"

# Make another scene on oseltamivir
orient OTV
scene F2, store

# Now you can press the functions keys F1 and F2 or run the commands:
scene F1, recall
scene F2, recall
```

Mini-movie (spin around a ligand and save frames):

1. Center on the ligand you want to showcase

```
orient ZMR
```

2. Build an 8-second (240 frames @30 fps) spin around the vertical (Y) axis Using the top menus: Movie -> Program -> Cameral Roll -> Y-Roll -> 8 seconds
3. Convert to MPEG and export
   Using the top menus: File -> Export Movie As -> MPEG -> (select size, format and destination)

---

10) Save your session and make a replayable `.pml` script

- **Session file** (keeps everything exactly as seen):

```
save neuraminidase_tutorial.pse
```

- **Replayable script** (a text file of commands you can re-run later):

```
log_open lesson_build.pml
# …do your steps…
log_close
```

You can also write your **own clean script** in a text editor and run it in PyMOL with:

```
@lesson.pml
```

11) Quick troubleshooting notes (students ask these a lot)

- **“I don’t see the ligand.”** Try `select LIG, organic` then `show sticks, LIG`. Some PDBs use different residue names.
- **“The dashed bonds look messy.”** Hide labels (`hide labels, hbonds`) and keep a **single** H-bond object per scene.
- **“Distances didn’t appear.”** Ensure you selected protein **N/O** atoms (not carbons) and used a reasonable cutoff (3.2–3.6 Å).
- **“My superposition looks weird.”** Align by backbone Cα only (`align <obj1> and name CA, <obj2> and name CA`) or try focusing on active-site residues with `pair_fit`.
- **“Residue numbers don’t match a paper figure.”** NA subtypes (N1, N2…) differ; published numbering (e.g., “N2 numbering”) may not match your PDB chain. Always **point at the 3D position**, not just the number.