

# Nature's BARDS

## Biotechnological, Artistic , Researchful Data Sonifiers



### TEAM ROLES:

Writer:

Fatima

Technical Writers:

Jacob, Rojina

Slide Design:

Samantha

SabDab Nano Curators:

Afsheen, Fatima, Thomas

Echinobase Explorers:

Afsheen, Thomas, Ei Kay

Code & Algorithms Finders:

Jacob, Rojina, Ei Kay

Music Liaison:

Jacob

Confusion & Chaos generator:

Thomas

# What problem are we trying to solve?

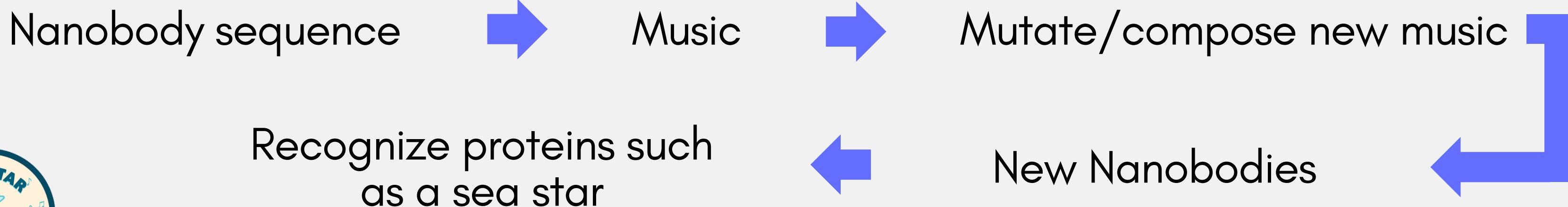


## GOAL:

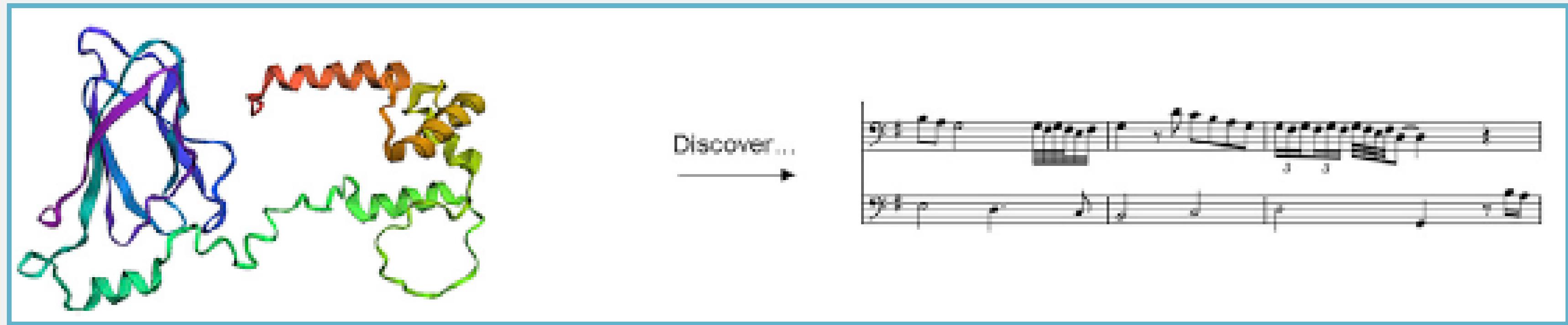
Exploring how music can help redesign existing nanobodies (Nbs) or design new Nbs capable of binding to target proteins

## QUESTION:

How can we use music to identify and design nanobodies?



**Inspiration from:** Buehler, M. J. (2023). Unsupervised cross-domain translation via deep learning and adversarial attention neural networks and application to music-inspired protein designs. *Patterns*, 4(3). <https://doi.org/10.1016/j.patter.2023.100692>



# What we plan to do by Thursday



01

## DATA/TOOLS

- Research existing algorithms, apps, & music/protein tools

02

## MUSIC

- Explore how to convert nanobody sequences to music
- Design potential new nanobodies

03

## DATABASES

- SabDab: nanobodies & structures
- Echinobase: for sea star protein targets

04

## FUTURE

- Can we use AI to predict binding
- Develop method to predict best musical composition for Nb-Ag binding

# Work Plan

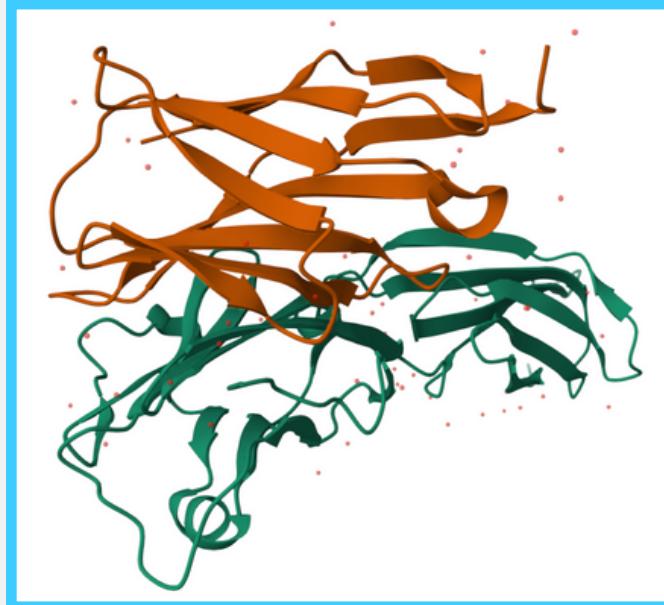
## GOAL:



Day	Key Tasks
Tues (8/5)	<ul style="list-style-type: none"><li>• Pull nanobody sequences from SabDab</li><li>• Grab sea star protein targets from Echinobase</li><li>• Pick sonification tool &amp; set up git repo</li></ul>
Wed (8/6)	<ul style="list-style-type: none"><li>• Convert first sequences into music</li><li>• Compose / mutate 3 tracks</li><li>• Document the conversion rules</li></ul>
Thurs (8/7)	<ul style="list-style-type: none"><li>• Translate the mutated music back into new sequences</li><li>• Run quick binding prediction (AlphaFold or docking)</li><li>• Results/Findings</li></ul>

# Nature's BARDS

Biotechnological, Artistic , Researchful Data  
Sonifiers



**5JDS, CHAIN B**  
nanobody from rcsb entry

# WEDNESDAY

**Team Leader:** Thomas

Fatima, Samantha, Jacob, Rojina, Afsheen, Ei Kay



Nature BARD's

# Progress



## CONVERTED PROTEIN SEQUENCE INTO SOUND/MUSIC

PYTHON CODE TOOK THE NANobody AMINO ACID SEQUENCE



MAPPED EACH AMINO ACID TO A FREQUENCY (PITCH)

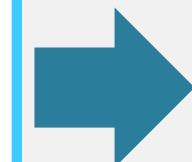
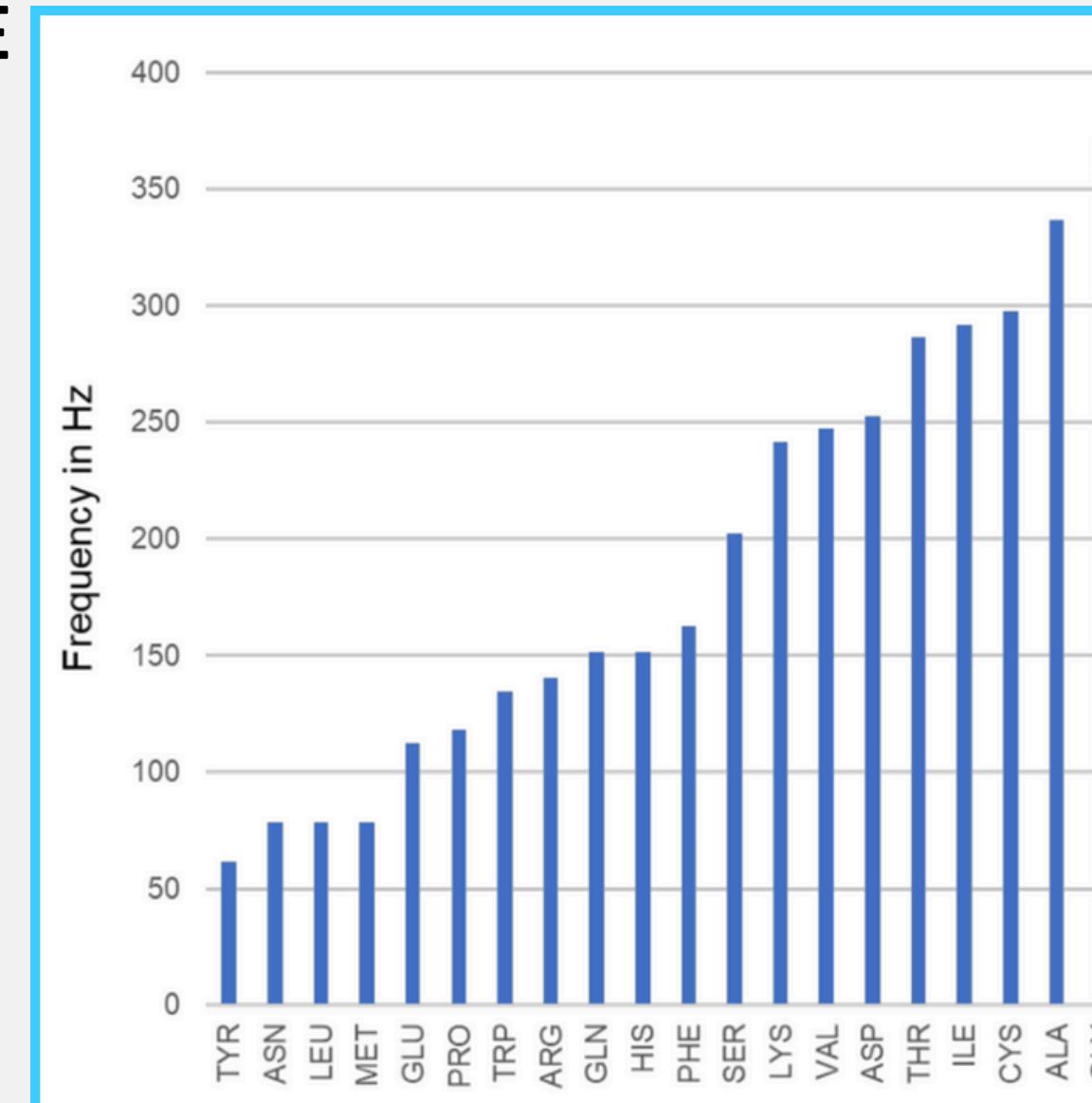


EXPORTED THAT AS A SOUND (WAV) OR MIDI FILE



# Method

- LOWEST NORMAL MODE VIBRATIONAL FREQUENCIES OF EACH AMINO ACID (FIGURE 2B IN YU ET AL., 2019)**
- KN035 NANOBODY SEQUENCE FROM CHAIN B PLAYED AS MELODY**



NOTE	Hz	Hz	NOTE	Hz
A0	27.5000	146.832	G5	783.991
A# 0 or Bb0	29.1352	155.563	G5	783.609
B0	30.8677	164.814	A # or Ab5	830.000
C1	32.6722	184.977	E3	155.328
C# 1 or DDb1	34.6578	34.6578	D1	987.767
D1	48.2693	207.652	G# or Db6	1396.50
G# 1 or Abb1	46.2493	271.626	A3	1569.91
A1	48.9994	233.082	D6	1396.51
A# 0 or Ab1	55.0000	263.082	A # or Bb6	1244.51
A2	41.2034	233.082	E6	1318.51
D# 1 or Bb3	43.6535	283.082	C7	2093.00
D1	48.9994	265.626	C# or Db7	2219.46
G1	48.9994	339.811	G7	2630.02
C8	4186.01	4186.01	C8	4186.01

**HZ TO NOTE REFERENCE CHART**

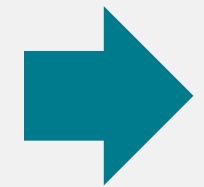
**MAPPING: AMINO ACID → FREQUENCY (Hz)**

Yu, C-H, et al. (2019). A Self-Consistent Sonification Method to Translate Amino Acid Sequences into Musical Compositions and Application in Protein Design Using Artificial Intelligence." ACS Nano, 13(6), 7471–7482. <https://drive.google.com/file/d/1rJk4AqWC1nzSYSVIQHIOXPD9ONLDFggA/view?usp=sharing>



# Demo

Nanobody



Music

PLAY KN035 MIDI FILE

(FIRST 52 AMINO  
ACIDS ONLY)

LINK TO DOWNLOAD/  
PLAY RECORDING

7

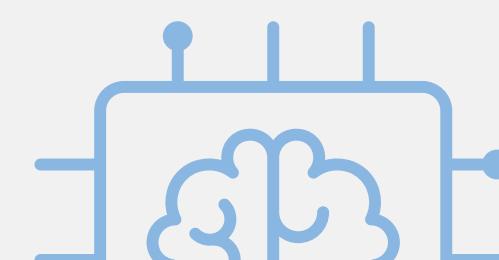
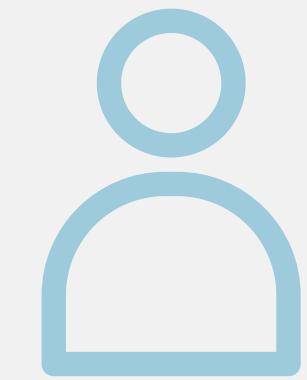
T G Q V Q L Q E S G G G L V Q P G G S L R L S C

A A S G K M S S R R C M A W F R Q A P G K E R E R V A K

# Challenges Faced

**CODE TO RECOGNIZE FR2 & CDR3  
REGIONS WITH MUSIC**

**TRAINING AN AI MODEL ON THESE  
MUSICAL PATTERNS**



# Nature's BARDS

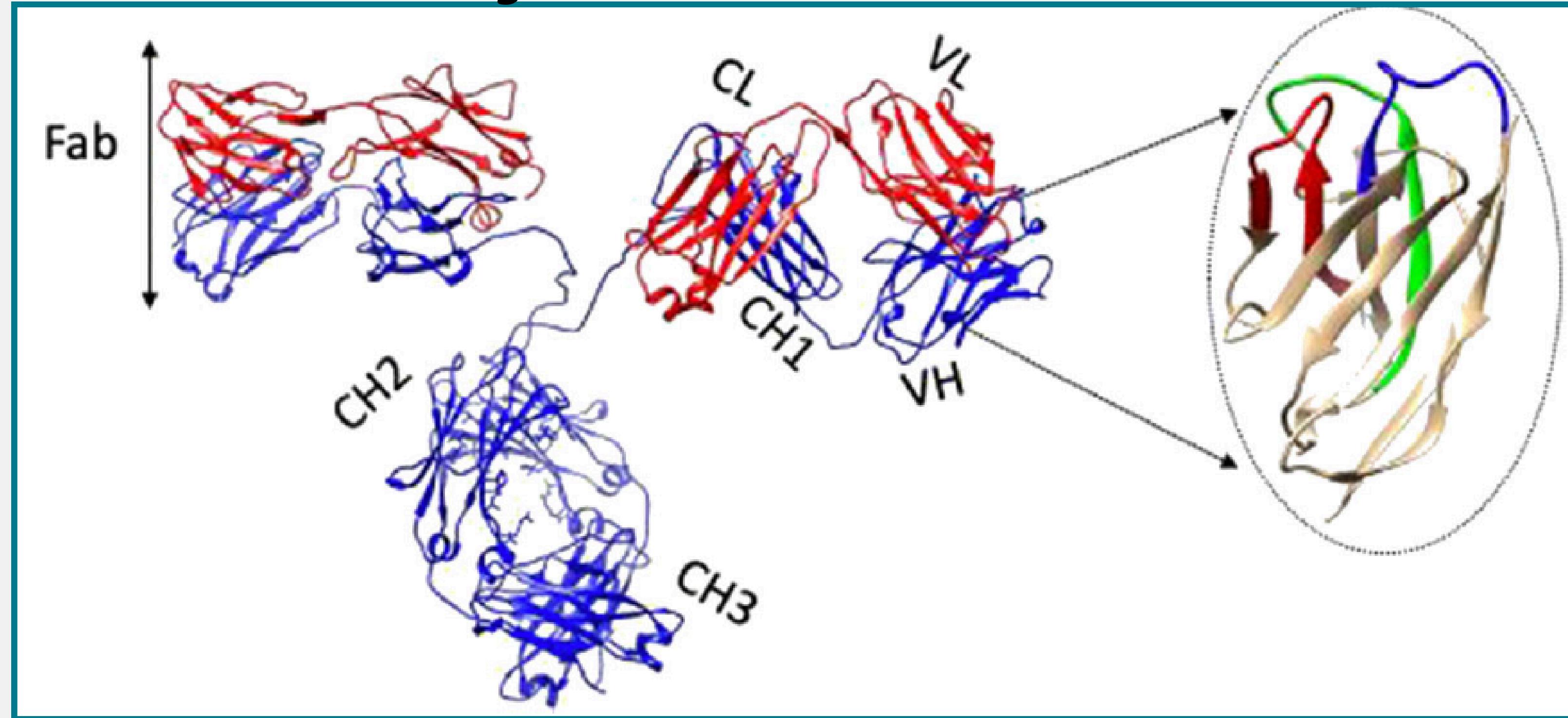
Biotechnological, Artistic , Researchful Data Sonifiers

## THURSDAY

**Team Leader:** Thomas Onorato

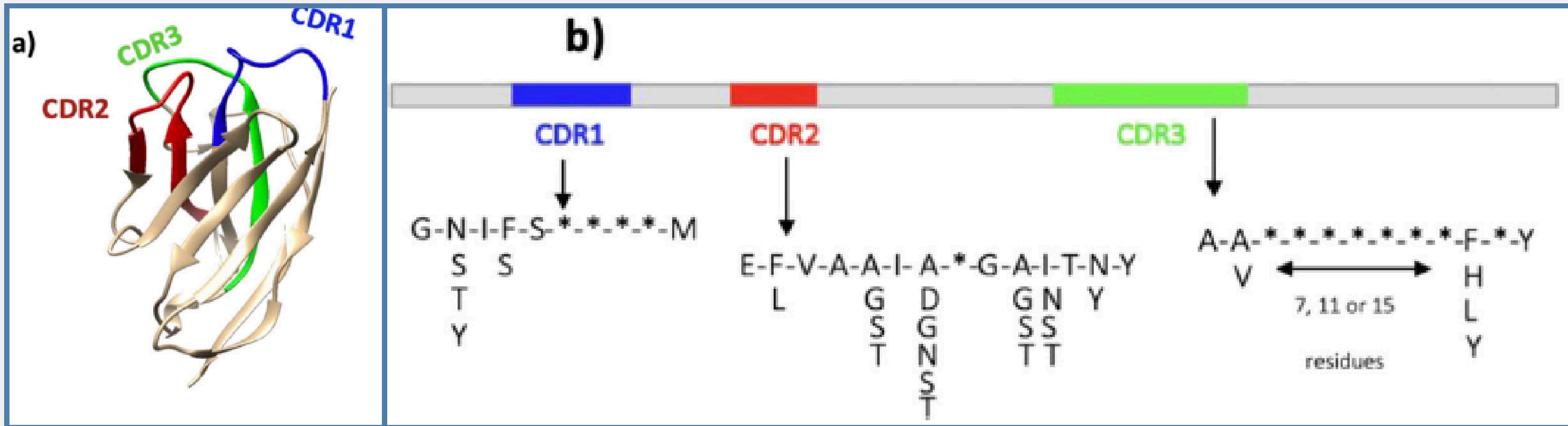
Fatima Ali, Samantha Barrera, Jacob Sagrans, Rojina Magar, Afsheen Rahman, Ei Kay Khaing Myo

# The structure, consensus sequence, and the hypervariable positions for *Lama glama* derived nanobodies



ModiBodies: A computational method for modifying nanobodies in nanobody-antigen complexes to improve binding affinity and specificity. Journal of Biological Physics (2020) 46:189–208 <https://doi.org/10.1007/s10867-020-09548-3>

# The structure, consensus sequence, and the hypervariable positions for *Lama glama* derived nanobodies



ModiBodies: A computational method for modifying nanobodies in nanobody-antigen complexes to improve binding affinity and specificity. Journal of Biological Physics (2020) 46:189–208 <https://doi.org/10.1007/s10867-020-09548-3>

**>2VYR\_2|Chains E, F, G, H, I, J, K, L|HUMAN SINGLE DOMAIN ANTIBODY|HOMO SAPIENS (9606).**

EVQLLESGGGLVQPGGSLRLSCAASGFTFEEYAMLWVRQAPGKGLEWVSGINAR  
GYTTYYADSVKGRFTISRDNSKNTLYLQMNSLRTEDTAVYYCAKPWYPFMASKG  
SEFDYWGQGTLTVSSAAALEIKRASQPELAPEEDVEHHHHHH

**>5JDS\_1|Chain A[auth B].|Nanobody|Camelidae (9835).**

TGQVQLQESGGGLVQPGGSLRLSCAASGKMSSRRRCMAWFRQAPGKERERVAKLLTSGSTYLA  
DSVKGRFTISRQNNAKSTVYLQMNSLKPEDTAMYCYCAADSFEDPTCTLVTSSGAFQYWGQGTQV  
TVSSGSMDPGGSHHHHHHHH

**>8DMO\_2|Chain B[auth H].|VH ab6|Homo sapiens (9606).**

EVQLVESGGGVVQPGRSLRLSCAASGFTFSSYAMHWVRQAPGKGLEWIGNIYHDGSTFYNP  
SLSLVTISRDDSTNTLYLQMNSLRAEDTAIYYCARVWLYGSGYMDWGKGT  
LTVSS

PDB Sequence Annotations in 3D: 2VYR

Structure of human MDM4 N-terminal domain bound to a single domain antibody

Chain: E HUMAN SINGLE DOMAIN ANTIBODY - Homo sapiens

Help Back

Residue

LINK TO  
DOWNLOAD  
VIDEO

CHAIN E: SLRLSCAASGFTFEEYAMLWVRQAPGKGLEWVSG

SECONDARY STRUCTURE: UNMODELED

EDS ATOMS

OCC WT AVG B

AVERAGE OCCUPANCY

ATOMIC CLASHES

RSRZ

RSR

RSCC

BINDING CHAIN A

BINDING CHAIN L

BURIED RESIDUES

DISULFIDE BRIDGE

HYDROPATHY

DISORDER

DISORDERED BINDING

CATH

SCOP

ECOD

ANTIBODY DOMAIN

20 25 30 35 40 45 50

HUMAN SINGLE DOMAIN ANTIBODY

2VYR | Model 1 | Instance ASM\_1 | E | ALA 24

# Code:

- Vibrational frequencies translated into sound
- Piano as background to ground composition

## Second Instrument:

- Finds 'AA'
- Look for 'Y' 7 characters after 'AA'
- If 'Y' is not found, end searching after 18 characters

```
sequence = "EVQLLESGGGLVQPGGSLRLSCAASGFTFEEYAMLWVRQAPGKGLEWSGINARGYTTYYADSVKGRFTISRDNSKNTLYLQMNSLRTEDTAVYYCA"

# Mapping each amino acid to a MIDI note (some are kinda random but whatever, it works)
```

```
# This one finds the AA spots that trigger the second instrument
def find_instrument2_segments(seq):
    segments = []
    i = 0
    while i < len(seq) - 1:
        if seq[i:i+2] == 'AA':
            start = i
            end = start + 7
            found_y = False
            for j in range(start + 7, min(start + 18, len(seq))):
                if seq[j] == 'Y':
                    end = j + 1
                    found_y = True
                    break
            if not found_y:
                end = min(start + 18, len(seq))
            segments.append((start, end))
            i = start + 1
        else:
            i += 1
    return segments
```

## Code:

- Searches for all hhh stretches in the sequence
- Uses them to trigger a drum sound 

```
# This one finds all HHH stretches and continues until it hits something that's not H
def find_hhh_segments(seq):
    segments = []
    i = 0
    while i < len(seq) - 2:
        if seq[i:i+3] == 'HHH':
            start = i
            end = i
            while end < len(seq) and seq[end] == 'H':
                end += 1
            segments.append((start, end))
            i = end
        else:
            i += 1
    return segments
```

## Code:

- Installation
- Create MIDI file with 3 tracks of the instruments (piano, violin, drums)
- Assign Amino Acid to a note

```
# If you're using this on Google Colab or something, install MIDIUtil
!pip install midiutil

from midiutil import MIDIFile

# Create MIDI with 3 tracks this time: piano, violin, and drums
mf = MIDIFile(3)
mf.addTempo(0, 0, 120)
mf.addTempo(1, 0, 120)
mf.addTempo(2, 0, 120)

# Piano (full sequence)
mf.addProgramChange(0, 0, 0, 0)    # Piano
# Violin (AA-triggered segments)
mf.addProgramChange(1, 1, 0, 40)   # Violin
# Drums are on channel 9 – we don't need to set an instrument, it's built-in

# --- Instrument 1: Piano playing full sequence ---
time = 0
aa_to_note = {
    'A': 60, 'C': 61, 'D': 62, 'E': 63, 'F': 64,
    'G': 65, 'H': 66, 'I': 67, 'K': 68, 'L': 69,
    'M': 70, 'N': 71, 'P': 72, 'Q': 73, 'R': 74,
    'S': 75, 'T': 76, 'V': 77, 'W': 78, 'Y': 79,
}
```

## Code:

### Instrument 2-Violin (AA to y)

- Looks for parts of the sequence that start with 'AA' and end with 'Y' and plays violin sound

### Instrument 3-Drums (HHH)

- Plays drum beat under HHH

```
# --- Instrument 2: Violin playing from AA to Y or 18 aa max ---
segments = find_instrument2_segments(sequence)

for start, end in segments:
    time = start
    for i in range(start, end):
        aa = sequence[i]
        if aa in aa_to_note:
            note = aa_to_note[aa]
            mf.addNote(track=1, channel=1, pitch=note, time=time, duration=1, volume=100)
            time += 1
```

```
# --- Instrument 3: Drums under HHH segments ---
hhh_segments = find_hhh_segments(sequence)

for start, end in hhh_segments:
    time = start
    for i in range(start, end):
        # Snare drum = 38
        mf.addNote(track=2, channel=9, pitch=38, time=time, duration=1, volume=120)
        time += 1

    # Save it
    with open("nanobody_with_drums.mid", "wb") as output_file:
        mf.writeFile(output_file)

    print("✓ MIDI file 'nanobody_with_drums.mid' created!")
```

# **SCORE & RECORDING OF 2VYR NANOBODY**

# GITHUB LINK TO DOWNLOAD VIDEO, .IPYNB CODE USED TO PRODUCE MIDI, PDF OF SCORE, MP3, & MUSIC XMLFILE OF 2VYR (CAN ADAPT TO OTHER NANOBODIES)

# SCORE MADE USING FREE MUSESORE SOFTWARE

# The End !

Thank You  
for  
Listening

Team Nature's BARDS

