Noah L. Schrick
1492657

PDB Lab. Structural informatics: Protein Databank (PDB) files, proteins, complexes, transcription binding, and DNA palindromes. When a region of DNA is transcribed into RNA, a protein (polymerase enzyme) unwinds the double-stranded DNA and transcribes it to RNA. The goal of this lab is to gain familiarity with DNA and protein structure and to better understand how a polymerase "knows" where along the DNA to begin transcription. DNA contains "special words" or promoter sequences that the enzyme can recognize and bind.

A. Exploring a PDB file. Go to rcsb.org and search for 1TGH. **1**. What is this structure? Download the pdb file and open it in a text editor. What is the letter of the first chain listed? (To help answer this question, look for the rows that start with an **ATOM** record in the pdb text file, and read the section of https://www.wwpdb.org/documentation/file-format-content/format33/sect9.html that describes the columns for Record: ATOM). What kind of biomolecule is this chain part of (protein or DNA)? What are the other chain ids (letters)? (Chains are separated by TER lines in the pdb file).

1. This is a TATA-box-binding protein found in homo sapiens.
The letter of the first chain listed is B.
This chain is part of DNA.
The other chain IDs are C and A.

B. Visualize a PDB 3D structure. Using the code below, install Rpdb, load the pdb file, and visualize the atomic structure. **2**. How many atoms are in the structure?

```
install.packages("Rpdb")
# On Mac requires installation of xquartz.org and restart Mac
library(Rpdb)
x<-read.pdb("1TGH.pdb")
natom(x)
visualize(x,type="l")
```

Use the code below to visualize the B and C chains.

```
# grab chains B and C
B_chain_pdb <- subset(x$atoms, x$atoms$chainid=="B")
C_chain_pdb <- subset(x$atoms, x$atoms$chainid=="C")
# remove water:
C_chain_pdb <- subset(C_chain_pdb,C_chain_pdb$resname!="HOH")
```

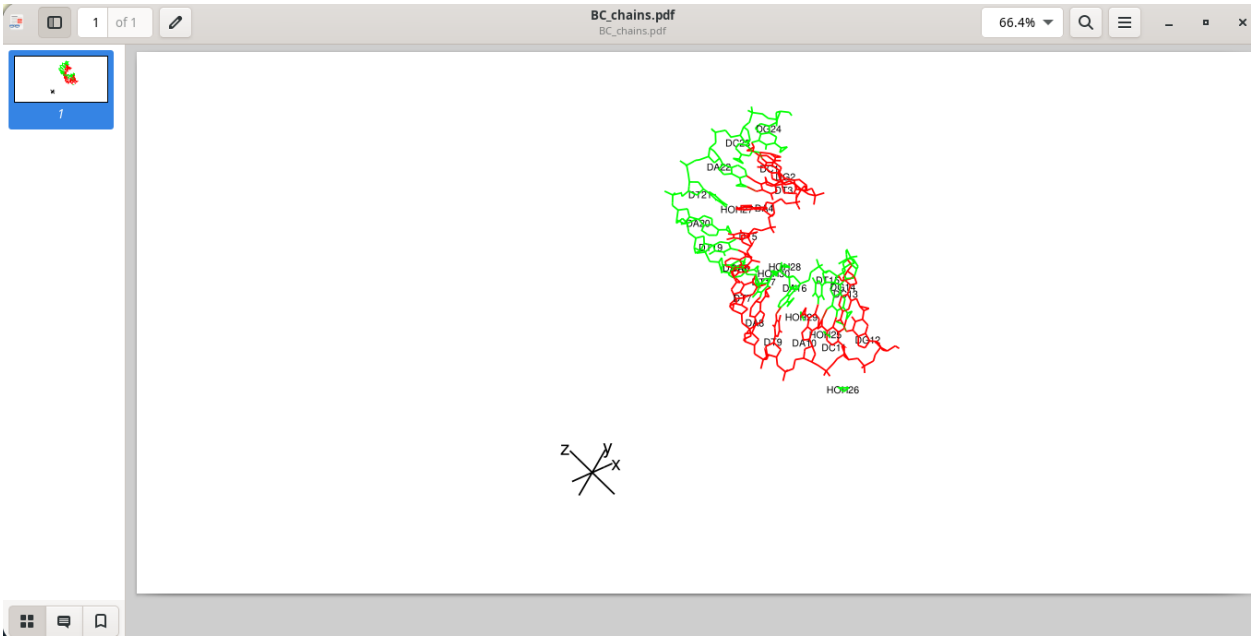
```
# visualize chains B and C, | is the OR logic operator
BC_chains_pdb <- subset(x$atoms, x$atoms$chainid=="B" |
                                        x$atoms$chainid=="C")
color.vec <-
c(rep("red",natom(B_chain_pdb)),rep("green",natom(C_chain_pdb)))

visualize(BC_chains_pdb,col=color.vec)
addResLab(BC_chains_pdb)
```

2. There are 2,355 atoms in the structure.

Rotate and zoom to get a desirable view of the chains, and then use the following command to save the structure as a pdf. **3.** Show the pdf. Biologically, what does the structure represent?

```
rgl.postscript("BC_chains.pdf","pdf",drawText=TRUE)
```

3. This represents the double-helix structure of DNA.



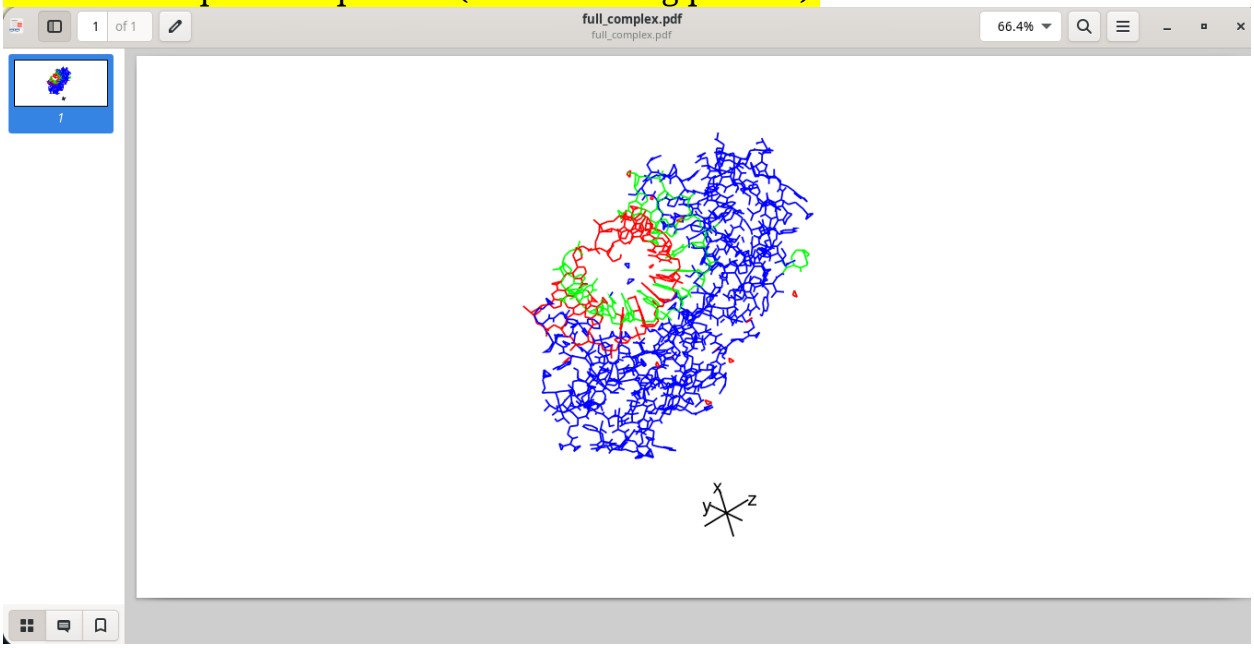
Use the code below to visualize the B-C and A chains.

```
# grab A chain
A_chain_pdb <- subset(x$atoms, x$atoms$chainid=="A")
# remove water
A_chain_pdb <- subset(A_chain_pdb, A_chain_pdb$resname!="HOH")

# visualize complex complex
BCA_chains_pdb <- subset(x$atoms, x$atoms$chainid=="B" |
                  x$atoms$chainid=="C" | x$atoms$chainid=="A")
BCA.color.vec <-
c(rep("red",natom(B_chain_pdb)),rep("green",natom(C_chain_pdb)),rep("b
lue",natom(A_chain_pdb)))

visualize(BCA_chains_pdb,col=BCA.color.vec)
```

Rotate and zoom to get a desirable view of the chains, and then use the following command to save the structure as a pdf. **4.** Show the pdf. What is chain A?

```
rgl.postscript("full_complex.pdf","pdf",drawText=TRUE)
```

4. Chain A is part of a protein (TATA binding protein).



C. <u>Primary structure and DNA Palindromes</u>. *Promoter sequences are typically reverse-complement palindromes, most famously the TATA box. Promoter sequences (DNA) are recognized by TATA-binding protein. Promoters often span a region of 200 nts (nucleotides) or more upstream of the transcriptional start site. When present, the TATA box occurs about 25 to 100 nt upstream of the start site. The TATA-binding protein (TBP) recognizes this TATA*

sequence and binds to it, creating a landmark that marks the start site of transcription. When the TBP binds to the DNA, it grabs the TATA sequence and bends it sharply.

Use the following to get the DNA sequence information from chain C. **5**. What looks odd about `C_chain_sequence_messy`? Show the sequence.

```
# get coordinates of C1' atoms of the C-chain DNA molecule
C_chain_pdb$resname
C_chain_resids<-unique(C_chain_pdb$resid)
C_chain_C1prime <- subset(C_chain_pdb, C_chain_pdb$elename=="C1'")

# get chain C DNA sequence
C_chain_sequence_messy <- C_chain_C1prime$resname
```

Use the following to fix this oddity and show the sequence.

```
C_chain_sequence <- paste(sapply(C_chain_sequence_messy,function(x)
{unlist(strsplit(x,""))[2]}),collapse = "")
```

5. > C_chain_sequence_messy
[1] "DC" "DG" "DT" "DA" "DT" "DA" "DT" "DA" "DT" "DA" "DC" "DG"
Rather than being one, joint sequence, it's a series of strings that all also begin with the letter "D", which should not be present.

6. Use the following code to install Biostrings and find palindromes.

```
library(BiocManager)
BiocManager::install("Biostrings")
library(Biostrings)

C_chain_DNAString <- DNAString(C_chain_sequence)
dna.pals <- findPalindromes(C_chain_DNAString, min.armlength=3,
                        max.looplength=5, max.mismatch = 0)
```

> dna.pals
Views on a 12-letter DNAString subject
subject: CGTATATATACG
views:
start end width
[1] 3 8 6 [TATATA]
[2] 1 12 12 [CGTATATATACG]
[3] 5 10 6 [TATATA]

D. Finding the binding site of the protein-DNA complex. The idea for finding the binding site is to find the closest residues between the protein and the DNA.

Earlier we grabbed the C1' atoms for the C chain DNA residues. Below, we get the 3D coordinates of the C1' atoms. **7.** What is the size (dimensions) of `C_chain_C1prime_coords`? What do the rows and columns represent?

```
C_chain_C1prime_coords <- coords(C_chain_C1prime)
```

7. > dim(C_chain_C1prime_coords)
[1] 12 3
The columns represent numeric vectors containing the first, second, and third coordinates (x, y, z). The rows are the IDs.

Similarly, we create an array of C-alpha coordinates of the A chain. **8.** What is the size of `A_chain_CA_coords`?

```
# get coordinates of CA atoms of the A-chain protein molecule
A_chain_sequence_3letter <- A_chain_pdb$resname
A_chain_resids<-unique(A_chain_pdb$resid)
A_chain_CA <- subset(A_chain_pdb, A_chain_pdb$elename=="CA")
A_chain_CA_coords <- coords(A_chain_CA)
```

8. > dim(A_chain_CA_coords)
[1] 180 3

Use the code below to create a distance matrix between all pairs of residues of the DNA chain and the protein chain. What is the size of the distance matrix? What chain/biomolecule do the rows and columns correspond to?

```
# epic one line function to compute a distance matrix between chains
pair.dist <- function(chain1,chain2)
{outer(1:nrow(chain1),1:nrow(chain2),Vectorize(function(i,j)
{dist(rbind(chain1[i,],chain2[j,]))}))}

prot2DNAdistMat <- pair.dist(A_chain_CA_coords,C_chain_C1prime_coords)

dim(prot2DNAdistMat)
```

8.5. > dim(prot2DNAdistMat)
[1] 180 12
The '180' correspond to the 180 inserts in the chain. The '12' correspond to the distance permutations between x,y,z coordinates in both chains.

9. What is the minimum distance between the two chains and what are the residues of the chains that are closest?

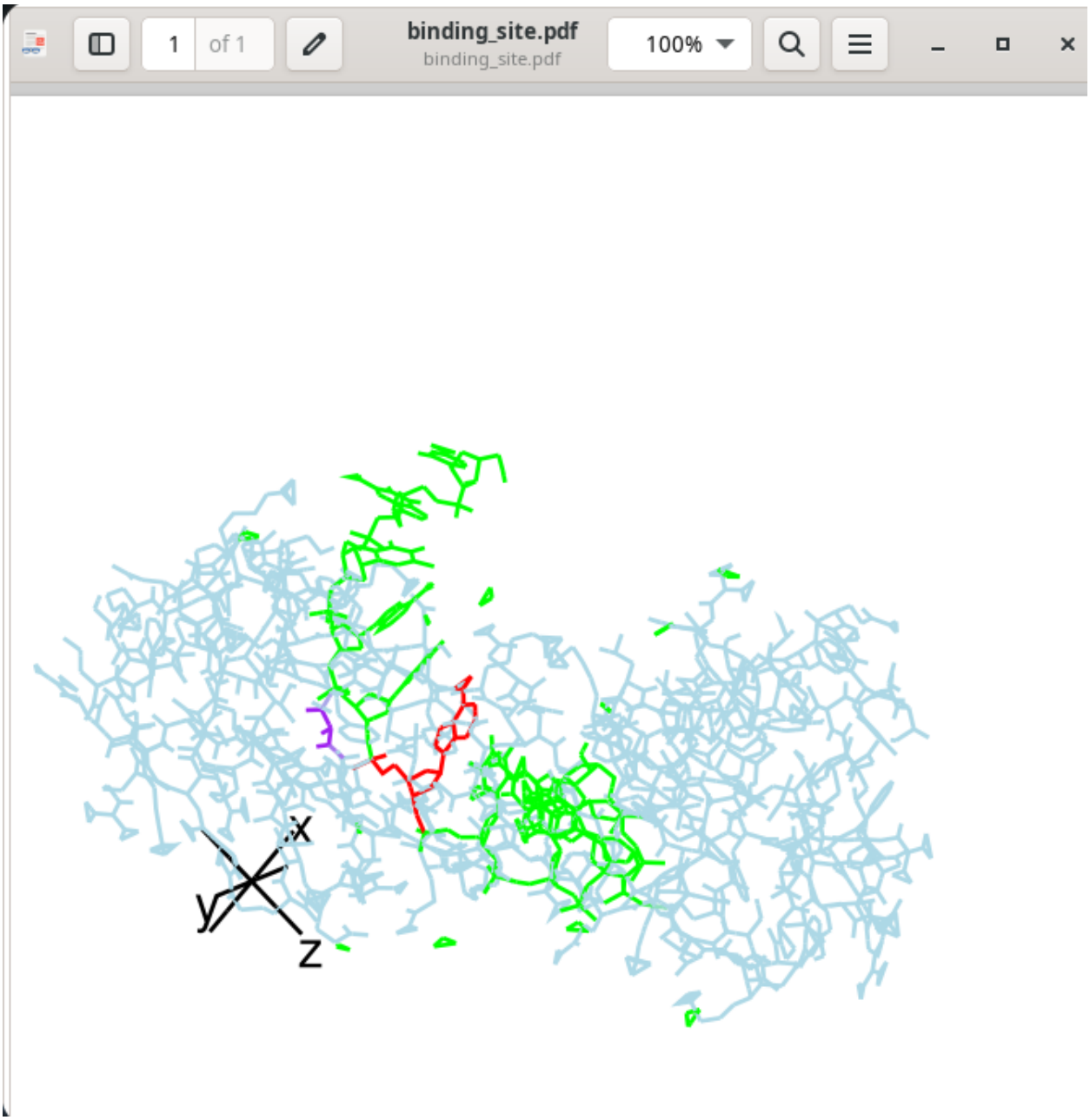
```
# ij location of min in current matrix (2-elt vector)
min_dist <- min(prot2DNAdistMat)
min_dist
min_ij <- which(prot2DNAdistMat == min_dist, arr.ind = TRUE)
min_ij
A_chain_sequence_3letter[min_ij[1]]  # closest A-chain residue
strsplit(C_chain_sequence,"")[[1]][min_ij[2]]  # closest C-chain
residue
```

Min dist is 4.94818
Closest A-chain residue: “GLN”
Closest C-chain residue: “A”

10. What is the relative sequence position of the DNA binding site with respect to the palindrome sequences? In the middle of the DNA? Skewed to one side?
In the middle

Use the `col` (color) option in `visualize` to color the two binding-site residues a different color, like purple.

```
# color binding residues

CA_chains_pdb <- subset(x$atoms, x$atoms$chainid == "C" | x$atoms$chainid ==
"A")

CA.color.vec <- c(rep("green", natom(C_chain_pdb)), rep("blue",
natom(A_chain_pdb)))

CA.color.vec[which(CA_chains_pdb$resid == min_ij[1])] <- "purple"

CA.color.vec[which(CA_chains_pdb$resid == min_ij[2])] <- "purple"

visualize(CA_chains_pdb, col=CA.color.vec)

rgl.postscript("binding_site.pdf", "pdf", drawText=TRUE)
```



E. Panlindromes in other organisms.

11. Download a fasta file of the DNA sequence for the bacteria gene “Bacillus amyloliquefaciens 16S” by searching https://www.ncbi.nlm.nih.gov/nuccore/

Use your code from lab1 to read in the fasta file and convert to a string of nucleotides (i.e., “ATACTG…”). Then modify code from D.6 above to find the

palindrome with `min.armlength=5`. Verify the longest palindrome is indeed a palindrome (type out the reverse and complement to verify they same in the report).

```
> toString(fasta.dna.pals)
[1] "GTGGAATTCCAC"

(Purposely left broken up instead of one joint string)
> fasta.dna.pals.rev
 [1] "C" "A" "C" "C" "T" "T" "A" "A" "G" "G" "T" "G"

(rc for “reverse complement)
> fasta.dna.pals.rc

[1] "GTGGAATTCCAC"

> toString(fasta.dna.pals) == fasta.dna.pals.rc

[1] TRUE
```

Optional. AlphaFold is a deep learning plus homology modeling software for structure prediction and complex prediction (docking of two structures). Paste the amino acid sequence for the TBP protein into this AlphaFold interface to see how the prediction looks (good I suspect because it’s probably in the pdb database). Run it again, but change the residue at the binding site. Note you can also predict the complex by separating the two sequences by a colon (:). Paste an image of the structure.

https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

Additional information for the curious.
https://www.bonvinlab.org/education/molmod_online/alphafold/