INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

BIOINFORMATICS

Práctica 3.

Introducción a Colaboratory

NOMBRE DEL ALUMNO: GARCÍA QUIROZ GUSTAVO IVAN

GRUPO: 7CV3

NOMBRE DEL PROFESOR: ROSAS TRIGUEROS JORGE LUIS

FECHA DE REALIZACIÓN DE LA PRÁCTICA: 26/02/2025

FECHA DE ENTREGA DEL REPORTE: 05/03/2025

Índice

1 Marco teórico. 1
1.1 Proteínas y sus estructuras 1
1.2 Formato PDB (Protein Data Bank) 1
1.3 Centroide y Centro de Masa 1
1.3.1 Centroide 1
1.3.2 Centro de Masa (COM) 1
1.4 APIs y bibliotecas para el análisis de proteínas 2
1.4.1 Biblioteca Requests 2
1.4.2 NumPy 2
1.4.3 Periodictable 2
1.5 PDB ID: 7FCD 2
2 Material y equipo. 4
2.1.1 Hardware 4
2.1.2 Software 4
3 Desarrollo de la práctica. 5
3.1 Descarga del archivo PDB 5
3.2 Análisis del archivo PDB 5
3.3 Cálculo del centroide 6
3.4 Cálculo del centro de masa (COM) 6
3.5 Resultados por cadena 7
3.6 Ejecución del código 8
4 Desafíos y soluciones 8
5 Conclusiones y recomendaciones. 9
6 Bibliografía. 10
7 Código 11

1 Marco teórico.

1.1 Proteínas y sus estructuras

Las proteínas son macromoléculas biológicas esenciales formadas por cadenas de aminoácidos. Su función biológica está directamente relacionada con su estructura tridimensional. Para estudiar estas estructuras, los científicos utilizan técnicas como la cristalografía de rayos X, la resonancia magnética nuclear (RMN) y, más recientemente, la criomicroscopía electrónica (cryo-EM).

1.2 Formato PDB (Protein Data Bank)

El Protein Data Bank (PDB) es un repositorio mundial de información estructural de macromoléculas biológicas. El formato PDB es un estándar para representar estructuras tridimensionales de proteínas y ácidos nucleicos. Cada archivo PDB contiene información detallada sobre las coordenadas atómicas, conexiones químicas, secuencias de aminoácidos, y otros datos experimentales.

En un archivo PDB, cada línea que comienza con "ATOM" representa un átomo en la estructura, incluyendo:

- Tipo de átomo
- Coordenadas (x, y, z) en angstroms
- Identificador de cadena
- Número de residuo
- Información sobre el factor de temperatura y la ocupación

1.3 Centroide y Centro de Masa

En el análisis estructural de proteínas, dos conceptos matemáticos son particularmente útiles:

1.3.1 Centroide

El centroide, o centro geométrico, representa el punto medio de todos los átomos en una estructura. Para un conjunto de n puntos con coordenadas (x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_n, y_n, z_n), el centroide se calcula como:

$$Centroide = \left(\sum\left(\frac{x}{n}\right), \sum\left(\frac{y}{n}\right), \sum\left(\frac{z}{n}\right)\right)$$

El centroide es útil para determinar la posición general de una cadena proteica o un dominio estructural.

1.3.2 Centro de Masa (COM)

A diferencia del centroide, el centro de masa tiene en cuenta las diferentes masas de los átomos. Se calcula como:

$$COM = \left(\frac{\sum m^1 x^1 + m^2 x^2 + \ldots + m_n x_n}{M}, \frac{\sum m^1 y^1 + m^2 y^2 + \ldots + m_n y_n}{M}, \frac{\sum m^1 z^1 + m^2 z^2 + \ldots + m_n z_n}{M}\right)$$

Donde M es la masa total ($M = m_1 + m_2 + ... + m_n$).

El COM es importante para estudiar las propiedades físicas de la proteína, incluyendo su comportamiento dinámico y sus interacciones con otras moléculas.

1.4 APIs y bibliotecas para el análisis de proteínas

Para procesar y analizar archivos PDB programáticamente, se utilizan diversas bibliotecas y APIs:

1.4.1 Biblioteca Requests

La biblioteca Requests de Python permite descargar archivos PDB directamente desde el repositorio RCSB PDB mediante peticiones HTTP.

1.4.2 NumPy

NumPy proporciona funciones eficientes para trabajar con arrays multidimensionales, lo que facilita los cálculos con coordenadas atómicas.

1.4.3 Periodictable

Esta biblioteca proporciona información sobre los elementos químicos, incluyendo sus masas atómicas, lo que es esencial para calcular el centro de masa con precisión.

1.5 PDB ID: 7FCD

El archivo PDB con identificador 7FCD corresponde a la estructura de una proteína específica obtenida experimentalmente. Analizando esta estructura mediante el cálculo del centroide y centro de masa de cada cadena, podemos obtener información valiosa sobre su organización espacial y propiedades físicas.

En la práctica actual, se desarrolla un script en Python que automatiza este análisis, permitiendo:

1. Descargar el archivo PDB desde el repositorio RCSB
2. Extraer las coordenadas atómicas y tipos de átomos por cadena
3. Calcular el centroide de cada cadena
4. Calcular el centro de masa considerando las diferentes masas atómicas
5. Presentar los resultados de manera organizada

Este tipo de análisis es fundamental en bioinformática estructural y proporciona información crucial para entender la función de las proteínas y su comportamiento en sistemas biológicos.

2 Material y equipo.

Para esta práctica se usaron las siguientes herramientas de software y hardware necesarias para realizar la práctica.

2.1.1 Hardware

- Computadora.

2.1.2 Software

- Google colaboratory.

3 Desarrollo de la práctica.

En esta práctica, se trabajó con la estructura de la proteína identificada con el código 7FCD en la base de datos de Proteínas (PDB). El objetivo principal fue descargar el archivo PDB correspondiente y calcular tanto el centroide como el centro de masa (COM) para cada una de las cadenas de la proteína. A continuación, se describe el proceso paso a paso:

3.1 Descarga del archivo PDB

El primer paso consistió en descargar el archivo PDB desde la base de datos RCSB PDB utilizando el código proporcionado. Para ello, se implementó la función download_pdb(pdb_id), que realiza una solicitud HTTP al servidor de RCSB y obtiene el contenido del archivo en formato de texto.

```python
def download_pdb(pdb_id):
    """Download a PDB file from the RCSB PDB database."""
    url = f"https://files.rcsb.org/download/{pdb_id}.pdb"
    response = requests.get(url)
    if response.status_code == 200:
        return response.text
    else:
        raise Exception(f"Failed to download PDB file. Status code: {response.status_code}")
```

Figura 1 Descarga del archivo PDB

3.2 Análisis del archivo PDB

Una vez descargado el archivo, se procedió a analizar su contenido utilizando la función parse_pdb(pdb_content). Esta función recorre cada línea del archivo PDB, identifica las líneas que comienzan con la palabra clave ATOM (que contienen información sobre los átomos de la proteína), y extrae las coordenadas tridimensionales (x, y, z) y los tipos de átomos para cada cadena de la proteína.

```python
def parse_pdb(pdb_content):
    """Parse PDB file and extract atom coordinates and atom types by chain."""
    chains = {}
    for line in pdb_content.split('\n'):
        if line.startswith('ATOM'):
            try:
                atom_type = line[76:78].strip()
                if not atom_type:  # If atom type is not available in columns 76-78
                    atom_type = line[12:16].strip()

                chain_id = line[21]
                x = float(line[30:38])
                y = float(line[38:46])
                z = float(line[46:54])

                if chain_id not in chains:
                    chains[chain_id] = {'coords': [], 'atom_types': []}

                chains[chain_id]['coords'].append([x, y, z])
                chains[chain_id]['atom_types'].append(atom_type)
            except (ValueError, IndexError):
                continue

    # Convert to numpy arrays for easier calculation
    for chain_id in chains:
        chains[chain_id]['coords'] = np.array(chains[chain_id]['coords'])

    return chains
```

Figura 2 Análisis del archivo PDB

3.3 Cálculo del centroide

El centroide de una cadena se define como el promedio de las coordenadas de todos los átomos que la componen. Para calcularlo, se implementó la función calculate_centroid(coords), que utiliza la función np.mean() de la librería NumPy para obtener el promedio de las coordenadas en cada eje (x, y, z).

```python
def calculate_centroid(coords):
    """Calculate the centroid (geometric center) of a set of coordinates."""
    return np.mean(coords, axis=0)
```

Figura 3 Cálculo del centroide

3.4 Cálculo del centro de masa (COM)

El centro de masa (COM) es un punto que representa la distribución de masa de una cadena. Para calcularlo, se implementó la función calculate_com(coords, atom_types), que toma en cuenta las coordenadas de los átomos y sus masas

atómicas. Las masas atómicas se obtuvieron utilizando la librería periodictable, que proporciona información sobre los elementos químicos. En caso de que no se encuentre la masa de un átomo específico, se asume una masa predeterminada (por ejemplo, la masa del carbono).

```python
def calculate_com(coords, atom_types):
    """Calculate the center of mass of a set of coordinates."""
    masses = []

    for atom in atom_types:
        # Remove digits from atom type (e.g., 'C1' -> 'C')
        element = ''.join([c for c in atom if not c.isdigit()]).strip()
        # Get only the first character if it's a two-letter element but not properly
        if len(element) > 2 or (len(element) == 2 and element[1].isupper()):
            element = element[0]

        try:
            # Try to get the mass from periodictable
            element_obj = getattr(periodictable, element)
            mass = element_obj.mass
        except (AttributeError, TypeError):
            # Default to carbon mass if element not found
            mass = 12.011

        masses.append(mass)

    masses = np.array(masses)
    total_mass = np.sum(masses)

    # Calculate center of mass
    com = np.sum(coords * masses[:, np.newaxis], axis=0) / total_mass

    return com, total_mass
```

Figura 4 Cálculo del centro de masa (COM)

3.5 Resultados por cadena

Finalmente, se calcularon el centroide y el centro de masa para cada cadena de la proteína utilizando la función analyze_pdb_structure(pdb_id). Los resultados se almacenaron en un diccionario que incluye:

- El número de átomos en la cadena.
- Las coordenadas del centroide.
- Las coordenadas del centro de masa.
- La masa total de la cadena.

3.6 Ejecución del código

El código se ejecutó para la proteína 7FCD, y los resultados se imprimieron en la consola. A continuación, se muestra un ejemplo de la salida obtenida:

```
Resultados por Cadena:
--------------------------------------------------------------------------------
Cadena A (7901 atomos):
  Centroide: [169.531, 143.499, 149.147]
  Centro de Masa: [169.525, 143.532, 149.116]
  Masa Total: 104219.45 unidades de masa atomica

Cadena B (8009 atomos):
  Centroide: [153.575, 174.635, 162.428]
  Centro de Masa: [153.571, 174.620, 162.366]
  Masa Total: 105652.37 unidades de masa atomica

Cadena C (8044 atomos):
  Centroide: [135.839, 147.080, 148.346]
  Centro de Masa: [135.883, 147.085, 148.321]
  Masa Total: 106128.60 unidades de masa atomica
```

4 Desafíos y soluciones

Durante la práctica, se enfrentaron algunos desafíos, como:

- Identificación correcta de los tipos de átomos: Algunos átomos en el archivo PDB no tenían un formato estándar, lo que dificultaba la extracción de su tipo. Para solucionarlo, se implementó un sistema de limpieza de cadenas que elimina dígitos y caracteres no deseados.
- Manejo de elementos químicos no estándar: En algunos casos, no se pudo obtener la masa atómica de ciertos elementos. Para evitar errores, se asignó una masa predeterminada (la del carbono).

5 Conclusiones y recomendaciones.

En esta práctica, se logró cumplir con el objetivo de calcular el centroide y el centro de masa (COM) para cada cadena de la proteína 7FCD. A través del uso de Python y librerías como NumPy y periodictable, se pudo automatizar la descarga, análisis y procesamiento de archivos PDB. Los resultados obtenidos proporcionan información valiosa sobre la distribución espacial y la masa de las cadenas de la proteína, lo cual es fundamental para estudios estructurales en bioinformática.

Además, se identificaron y resolvieron desafíos relacionados con la extracción de datos de archivos PDB, como la identificación correcta de tipos de átomos y el manejo de elementos químicos no estándar. Esto demuestra la importancia de implementar métodos robustos para el procesamiento de datos biológicos.

Se recomienda validar los resultados obtenidos comparándolos con herramientas bioinformáticas establecidas, como PyMOL o VMD, para asegurar la precisión de los cálculos.

Para futuras prácticas, se sugiere implementar un sistema más avanzado para identificar tipos de átomos, especialmente en casos donde el formato del archivo PDB no sigue los estándares.

podría optimizarse para manejar archivos PDB más grandes y complejos, utilizando técnicas de paralelización o procesamiento por lotes.

Sería interesante extender el análisis para incluir cálculos adicionales, como el radio de giro o la distancia entre cadenas, lo que permitiría un estudio más completo de la estructura proteica.

6 Bibliografía.

[1] H. M. Berman et al., "The Protein Data Bank," Nucleic Acids Research, vol. 28, no. 1, pp. 235–242, Jan. 2000. doi: 10.1093/nar/28.1.235.

[2] NumPy Developers, "NumPy: Fundamental package for scientific computing with Python," 2023. [Online]. Available: https://numpy.org.

[3] P. A. Keller, "Periodic Table of the Elements," 2023. [Online]. Available: https://pypi.org/project/periodictable/.

[4] Python Software Foundation, "Python Language Reference," 2023. [Online]. Available: https://www.python.org.

[5] RCSB PDB, "RCSB Protein Data Bank: Tools for Exploring Structural Biology," 2023. [Online]. Available: https://www.rcsb.org.

[6] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, May 2007. doi: 10.1109/MCSE.2007.55.

7 Código

```
import numpy as np
import requests
from io import StringIO
import periodictable


def download_pdb(pdb_id):
    """Download a PDB file from the RCSB PDB database."""
    url = f"https://files.rcsb.org/download/{pdb_id}.pdb"
    response = requests.get(url)
    if response.status_code == 200:
        return response.text
    else:
        raise Exception(f"Failed to download PDB file. Status code: 
{response.status_code}")


def parse_pdb(pdb_content):
    """Parse PDB file and extract atom coordinates and atom types by chain."""
    chains = {}
    for line in pdb_content.split('\n'):
        if line.startswith('ATOM'):
            try:
                atom_type = line[76:78].strip()
                if not atom_type:  # If atom type is not available in columns 76-78
                    atom_type = line[12:16].strip()


                chain_id = line[21]
                x = float(line[30:38])
```

```
                y = float(line[38:46])
                z = float(line[46:54])

                if chain_id not in chains:
                    chains[chain_id] = {'coords': [], 'atom_types': []}

                chains[chain_id]['coords'].append([x, y, z])
                chains[chain_id]['atom_types'].append(atom_type)
            except (ValueError, IndexError):
                continue

    # Convert to numpy arrays for easier calculation
    for chain_id in chains:
        chains[chain_id]['coords'] = np.array(chains[chain_id]['coords'])

    return chains

def calculate_centroid(coords):
    """Calculate the centroid (geometric center) of a set of coordinates."""
    return np.mean(coords, axis=0)

def calculate_com(coords, atom_types):
    """Calculate the center of mass of a set of coordinates."""
    masses = []

    for atom in atom_types:
        # Remove digits from atom type (e.g., 'C1' -> 'C')
        element = ''.join([c for c in atom if not c.isdigit()]).strip()
```

```
        # Get only the first character if it's a two-letter element but not properly formatted
        if len(element) > 2 or (len(element) == 2 and element[1].isupper()):
            element = element[0]

        try:
            # Try to get the mass from periodictable
            element_obj = getattr(periodictable, element)
            mass = element_obj.mass
        except (AttributeError, TypeError):
            # Default to carbon mass if element not found
            mass = 12.011

        masses.append(mass)

    masses = np.array(masses)
    total_mass = np.sum(masses)

    # Calculate center of mass
    com = np.sum(coords * masses[:, np.newaxis], axis=0) / total_mass

    return com, total_mass

def analyze_pdb_structure(pdb_id):
    """Analyze the PDB structure to find centroids and centers of mass for each chain."""
    print(f"Analizando PDB: {pdb_id}")

    # Download PDB file
```

```
    pdb_content = download_pdb(pdb_id)

    # Parse PDB file
    chains = parse_pdb(pdb_content)

    # Calculate centroid and COM for each chain
    results = {}

    for chain_id, data in chains.items():
        coords = data['coords']
        atom_types = data['atom_types']

        centroid = calculate_centroid(coords)
        com, total_mass = calculate_com(coords, atom_types)

        results[chain_id] = {
            'num_atoms': len(coords),
            'centroid': centroid,
            'com': com,
            'total_mass': total_mass
        }

    return results

# Main execution
if __name__ == "__main__":
    # Analyze the specified PDB structure
    pdb_id = "7FCD"
```

```
    results = analyze_pdb_structure(pdb_id)


    # Print results
    print("\nResultados por Cadena:")
    print("-" * 80)
    for chain_id, data in results.items():
        print(f"Cadena {chain_id} ({data['num_atoms']} atomos):")
        print(f"  Centroide: [{data['centroid'][0]:.3f}, {data['centroid'][1]:.3f}, {data['centroid'][2]:.3f}]")
        print(f"  Centro de Masa: [{data['com'][0]:.3f}, {data['com'][1]:.3f}, {data['com'][2]:.3f}]")
        print(f"  Masa Total: {data['total_mass']:.2f} unidades de masa atomica")
        print()
```