

```

# Función principal
def main():
    archivo_entrada = "Vibrio_Cholerae_Genome.fna"
    archivo_salida_cromosomas = "chromosomes_analysis.txt"
    archivo_salida_plásmidos = "plasmids_analysis.txt"

    resultados_cromosomas = []
    resultados_plásmidos = []

    try:
        with open(archivo_entrada, "r") as archivo:
            secuencia_actual = ""
            nombre_entrada_actual = ""

            for linea in archivo:
                if linea.startswith(">"):
                    if secuencia_actual:
                        secuencia_filtrada = secuencia_actual.upper() # Convertir a
mayúsculas

                        entrada = {
                            "nombre": nombre_entrada_actual,
                            "regiones": analizar_secuencia(secuencia_filtrada),
                        }
                        if "plasmid" in nombre_entrada_actual.lower():
                            resultados_plásmidos.append(entrada)
                        else:
                            resultados_cromosomas.append(entrada)
                        secuencia_actual = ""
                        nombre_entrada_actual = linea.strip()
                    else:
                        secuencia_actual += linea.strip()

            if secuencia_actual:
                secuencia_filtrada = secuencia_actual.upper()
                entrada = {
                    "nombre": nombre_entrada_actual,
                    "regiones": analizar_secuencia(secuencia_filtrada),
                }
                if "plasmid" in nombre_entrada_actual.lower():
                    resultados_plásmidos.append(entrada)
                else:
                    resultados_cromosomas.append(entrada)

    guardar_resultados_en_archivo(resultados_cromosomas, archivo_salida_cromosomas)
    guardar_resultados_en_archivo(resultados_plásmidos, archivo_salida_plásmidos)

    crear_histograma(
        resultados_cromosomas,
        "Análisis de Cromosomas",
        "Regiones",
        "Porcentaje",
        "chromosomes_histogram.pdf",
    )
    crear_histograma(
        resultados_plásmidos,
        "Análisis de Plásmidos",
        "Regiones",
        "Porcentaje",
        "plasmids_histogram.pdf",
    )

    print(f"Análisis completado. Resultados guardados en {archivo_salida_cromosomas}
y {archivo_salida_plásmidos}")

```

```

except FileNotFoundError:
    print("Error: El archivo de entrada no fue encontrado. Por favor verifica la ruta y vuelve a intentarlo.")
except Exception as e:
    print(f"Ocurrió un error: {e}")

```

La función `main` es el núcleo del programa, ya que organiza y ejecuta las tareas principales: leer un archivo de genoma, analizar las secuencias de cromosomas y plásmidos, guardar los resultados en archivos de texto y generar histogramas en formato PDF. A continuación, se explica su estructura:

1. Archivos de entrada y salida

```

archivo_entrada = "Vibrio_Cholerae_Genome.fna"
archivo_salida_cromosomas = "chromosomes_analysis.txt"
archivo_salida_plásmidos = "plasmids_analysis.txt"

```

- `archivo_entrada`: Nombre del archivo FASTA que contiene las secuencias de ADN.
- `archivo_salida_cromosomas`: Archivo de texto donde se guardan los análisis de los cromosomas.
- `archivo_salida_plásmidos`: Archivo de texto donde se guardan los análisis de los plásmidos.

2. Inicialización de listas de resultados

```

resultados_cromosomas = []
resultados_plásmidos = []

```

- Estas listas almacenarán los datos procesados de las secuencias, organizados por cromosomas y plásmidos.

3. Lectura del archivo FASTA

```

with open(archivo_entrada, "r") as archivo:
    secuencia_actual = ""
    nombre_entrada_actual = ""

    for linea in archivo:
        if linea.startswith(">"):
            if secuencia_actual:
                secuencia_filtrada = secuencia_actual.upper() # Convertir a mayúsculas
                entrada = {
                    "nombre": nombre_entrada_actual,
                    "regiones": analizar_secuencia(secuencia_filtrada),
                }
                if "plasmid" in nombre_entrada_actual.lower():
                    resultados_plásmidos.append(entrada)
                else:
                    resultados_cromosomas.append(entrada)
                secuencia_actual = ""
                nombre_entrada_actual = linea.strip()
            else:
                secuencia_actual += linea.strip()

```

- `with open`: Abre el archivo de entrada en modo lectura.
- `linea.startswith(">")`: Detecta las líneas de encabezado (indicadas por >), que contienen el nombre de la secuencia.
- Si se encuentra un encabezado mientras ya se está leyendo una secuencia, se procesa la secuencia actual:

- Se convierte a mayúsculas.
- Se analiza con la función `analizar_secuencia`.
- Se clasifica como cromosoma o plásmido según el texto del encabezado.

4. Procesar la última secuencia

```
if secuencia_actual:
    secuencia_filtrada = secuencia_actual.upper()
    entrada = {
        "nombre": nombre_entrada_actual,
        "regiones": analizar_secuencia(secuencia_filtrada),
    }
    if "plasmid" in nombre_entrada_actual.lower():
        resultados_plásmidos.append(entrada)
    else:
        resultados_cromosomas.append(entrada)
```

- Una vez terminado el bucle, se asegura de procesar la última secuencia leída, si existe.

5. Guardar resultados en archivos

```
guardar_resultados_en_archivo(resultados_cromosomas, archivo_salida_cromosomas)
guardar_resultados_en_archivo(resultados_plásmidos, archivo_salida_plásmidos)
```

- Se guardan los resultados de cromosomas y plásmidos en sus respectivos archivos usando la función `guardar_resultados_en_archivo`.

6. Generar histogramas

```
crear_histograma(
    resultados_cromosomas,
    "Análisis de Cromosomas",
    "Regiones",
    "Porcentaje",
    "chromosomes_histogram.pdf",
)
crear_histograma(
    resultados_plásmidos,
    "Análisis de Plásmidos",
    "Regiones",
    "Porcentaje",
    "plasmids_histogram.pdf",
)
```

- Se generan gráficos de barras con la función `crear_histograma` y se guardan en archivos PDF, diferenciando entre cromosomas y plásmidos.

7. Gestión de errores

```
except FileNotFoundError:
    print("Error: El archivo de entrada no fue encontrado. Por favor verifica la ruta y vuelve a intentarlo.")
except Exception as e:
    print(f"Ocurrió un error: {e}")
```

- **FileNotFoundError**: Captura errores si el archivo de entrada no se encuentra.

- **Exception:** Captura otros errores y muestra un mensaje indicando el problema.

8. Mensaje final

```
print(f"Análisis completado. Resultados guardados en {archivo_salida_cromosomas} y  
{archivo_salida_plásmidos}")
```

- Una vez completado todo el proceso, informa al usuario que el análisis se ha realizado correctamente y dónde están los resultados.

Ejemplo de Ejecución

Si ejecutas la función con un archivo FASTA llamado `Vibrio_Cholerae_Genome.fna`, obtendrás:

- `chromosomes_analysis.txt`: Contendrá los análisis de las regiones de los cromosomas.
- `plasmids_analysis.txt`: Contendrá los análisis de los plásmidos.
- `chromosomes_histogram.pdf`: Histogramas de los cromosomas.
- `plasmids_histogram.pdf`: Histogramas de los plásmidos.