

README

Archivos necesarios:

- archivo Genbank de la secuencia de interés → `secuencia.gb`
- `pipeline.sh`
 - `ejercicio_1.py`
 - `ejercicio_2.py`
 - `ejercicio_3.py`
 - `ejercicio_5.py`
- `emboss.sh` (corresponde a la resolución del ejercicio 4)
 - `dominios.py`

Pasos para ejecución:

- otorgar permisos sobre los archivos `pipeline.sh` y `emboss.sh` para poder ejecutarlos
- para ejecutar el archivo elegido escribir en el terminal:

```
./pipeline.sh secuencia.gb  
./emboss.sh secuencia.gb
```

Archivos generados:

- `pipeline.sh`:
 - Ejercicio 1: devuelve un archivo `ORFs.fasta` con n ORF encontrados
 - Ejercicio 2: devuelve n archivos, uno por cada ORF, con los resultados de BLAST
 - Ejercicio 3: devuelve un archivo `alignement.fasta` con el alineamiento múltiple
 - Ejercicio 5: devuelve un archivo `primers.fasta` con el n primers generados
- `emboss.sh`:
 - Devuelve un archivo `orf_secuencias.fasta` con los ORF encontrados con emboss
 - Devuelve un carpeta que contiene un archivo `.json` por cada secuencia que contiene el resultado de la búsqueda de prosite