**Proyecto Final: Análisis de Rendimiento de Dotplot Secuencial vs. Paralelización**

**Descripción General**

El objetivo de este proyecto es implementar y analizar el rendimiento de tres formas de realizar un dotplot, una técnica comúnmente utilizada en bioinformática para comparar secuencias de ADN o proteínas.

Los estudiantes implementarán una versión secuencial, una versión con hilos, una versión paralela utilizando la biblioteca *multiprocessing* de Python, una versión paralela utilizando *mpi4py y una versión usando pyCuda*. Compararán el rendimiento de estas tres implementaciones utilizando varias métricas y visualizarán sus resultados en un informe estilo artículo científico en formato IEEE.

**Requisitos del Proyecto**

**Implementación**

**1.** **Aplicación de línea de comandos para dotplot:** Los estudiantes deben crear una aplicación de línea de comandos que pueda tomar dos secuencias (en formato de archivos FASTA) como entrada y producir un dotplot de las secuencias. La aplicación debe ser capaz de ejecutar el dotplot de tres maneras: secuencialmente, utilizando *multiprocessing*, utilizando *mpi4py y utilizando pyCuda*.

**2.** **Función de filtrado de imagen:**Los estudiantes deben implementar una función paralela para realizar un filtrado de la imagen generada y detectar las líneas diagonales. Esta función deberá optimizarse para el rendimiento.

**Análisis de rendimiento**

Los estudiantes deben calcular y analizar las siguientes métricas para cada implementación:

1. Tiempos de ejecución totales y parciales (porción paralelizable).

2. Tiempo de carga de los datos y de generación de la imagen.

3. Tiempo muerto (tiempo no empleado en la ejecución del problema).

4. Aceleración y eficiencia.

5. Escalabilidad.

Se deben realizar gráficas de desempeño, aceleración, eficiencia y escalabilidad para facilitar la comparación entre las tres implementaciones.

**Datos de prueba**

Los estudiantes deben probar su aplicación con dos archivos FASTA que contienen los cromosomas X del Homo sapiens y del chimpancé. Estos archivos proporcionarán un buen conjunto de datos de prueba debido a su tamaño y las expectativas de similitud entre las secuencias.

**Informe**

Los estudiantes deben preparar un informe en el estilo de un artículo científico de IEEE que presenta los resultados de su análisis de rendimiento. El informe debe incluir una introducción al problema y a las implementaciones utilizadas, una descripción detallada de las métricas de rendimiento y una discusión de los resultados. Las gráficas deben utilizarse para ilustrar las diferencias en el rendimiento entre las tres implementaciones.

**Entrega**

El proyecto final debe ser entregado en un repositorio de GitHub que incluya el código fuente de la aplicación, los archivos de datos de prueba, y el informe en formato IEEE. El repositorio también debe incluir un archivo README que describa cómo ejecutar la aplicación y cualquier otra información que sea útil para entender el proyecto.

**Evaluación**

El proyecto será evaluado según los siguientes criterios:

- Implementación del dotplot y la función de filtrado de imagen.

- Cobertura y precisión del análisis de rendimiento.

- Claridad y calidad del informe.

- Eficacia de la aplicación de línea de comandos (facilidad de uso, manejo de errores, etc.).

**Recursos:**

* Carpeta con los scripts y recursos vistos en clase : [Carpeta de la Sesión en clase](https://drive.google.com/drive/folders/1k6PdL7ofpSIedq-DzgDBCnkkKgHZsz7_?usp=sharing)
* Archivos con los que deben probar la aplicación: [Insumos (archivos para ejecutar el programa)](https://drive.google.com/drive/folders/1LczMK2AY78Eg0DeHSSfDOq4Xchwz4AaS?usp=drive_link)
* Artículo científico de G-SAIP, un programa para realizar un dotplot: [Paper de G-SAIP](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9871978/pdf/10.1177_11769343221150585.pdf)
* Repositorio de G-SAIP (para que se guíen en la elaboración de la aplicación por línea de comandos): [Repo: G-SAIP: Graphical sequence alignment with HPC](https://github.com/simonorozcoarias/G-SAIP)

Recuerden que la aplicación debe ser ejecutable por línea de comandos. Ej:  
  
mpirun -n 4 python miAplicacionMPI.py - -file1=salmonella.fna - -file2=E.coli.fna --thres=0.8 -- output=miDotplot.py

***¡¡Muchos éxitos!!***