INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA ACADÉMICA DE INGENIERÍA
EN COMPUTADORES

CURSO: CE-4302 ARQUITECTURA DE COMPUTADORES II

Paralelización de Algoritmos de Alineación de Secuencias por medio de Arquitecturas SIMD (extensiones SIMD y GPU)

Profesor:
Ing. Jeferson González Gómez, M.Sc

Estudiante:

Esteban A. Sanabria Villalobos 2015070913

26 de mayo, 2019

***Resumen*—El presente documento tiene como objetivo el de explicar a detalle la investigación a ser desarrollada como proyecto final del curso de Arquitectura de Computadores II, de forma tal que se plantea realizar la combinación del área de Ing. en Computadores con el estudio de la Biología moderna, más específicamente en el campo de la Biotecnología. De forma que los algoritmos y técnicas básicas que se utilizan hoy en día para el estudio de secuencias de ADN y proteínas se pueda ver beneficiado por un posible aumento en el desempeño, a la hora de procesar los grandes volúmenes de datos recolectados de los diferentes organismos. El documento, explicara a detalle la idea general de la investigación, su motivo de ser; las bases teóricas que están por detrás de los procedimientos y algoritmos a ser utilizados, los objetivos de la presente investigación en conjunto con la metodología de desarrollo y el plan de acción para el desarrollo del mismo. Finalmente, se brindaran detalles del cronograma de trabajo, presupuesto, así como los planes de divulgación y transferencia de la tecnología a ser desarrollada.**

***Palabras Clave*—ADN, Algoritmo, Alineamiento, Alineamiento Local, Alineamiento Global, Benchmark, CUDA, GPU, Paralelización, Secuencias, SIMD.**

I. Descripción general del proyecto

Primeramente, la idea detrás de este proyecto surge de parte del interés de llevar los conocimientos del área de Ing. en Computadores al área de la Biotecnología, porque si bien la primera se define como ”La disciplina que incorpora la ciencia y la tecnología del diseño, la construcción, implementación y el mantenimiento de los componentes tanto de software como de hardware de los sistemas informáticos modernos y de los equipos controlados por computadora”, la segunda por su parte, es .[el] área de la biología que utiliza la información de los procesos en seres vivos para la explotación de procesos biológicos para fines industriales y otros, específicamente la manipulación genética de microorganismos para la producción de antibióticos, hormonas, etc.”; de forma que ambas áreas resultan estar separadas la una de la otra, pero no por eso significa que la unión de estas sea imposible y es ahí donde entra el concepto de Bioinformática, que es ”la combinación de la informática y la biología para la recopilación, clasificación, almacenamiento y análisis de información bioquímica y biológica utilizando computadoras en el modelado y resolución de los problemas en organismos y la naturaleza, especialmente aplicada en la genética molecular y genómica”, siendo esta el área que se desea beneficiar por medio de la presente propuesta.

De forma tal que, el proyecto consistirá de la realización de un análisis de consumo de tiempo y memoria en cuanto al rendimiento de los algoritmos utilizados en el alineamiento de secuencias de gran extensión, las cuales en nuestro caso procederán de la base de datos .[A]mtDB”(Ancient mitochondrial DNA database), la cual nos proveerá de una amplia gama de secuencias de ADN proveniente de multiples organismos a través de la historia; las secuencias serán procesadas por dos diferentes métodos de alineamiento, alineamiento Global y Local. Dichos algoritmos serán planteados en tres diferentes versiones, entre ellas: secuencial, optimización con extensiones SIMD y optimización para su procesamiento en GPU’s (CUDA). Así mismo, se realizará el diseño e implementación de un Benchmark, el cual se encargara de ejecutar múltiples casos de prueba con secuencias de diversas longitudes y contra cada uno de los algoritmos de alineamiento mencionados en cada una de sus versiones, para de esa forma recompilar, promediar y brindar un resultado representativo de si es posible mejorar el desempeño y productividad de procesos bioinformáticos que utilicen como parte de su desarrollo este tipo de procesos de alineamiento para la obtención de información relevante en el área de la biología genética o molecular.

A todo esto, el problema que se desea resolver radica en la cantidad de tiempo y memoria que se requiere a la hora de procesar grandes volúmenes de datos presentes en las diversas áreas de estudio y aplicación de la Bioinformática, un ejemplo en el estudio de biología molecular, donde se realiza el proceso de estudio de la replicación, transcripción y traducción de las cadenas de ADN a ARN a proteínas. Tal que, una secuencia de ADN puede llegar a contener el código genético de construcción de miles de proteínas, tal que por medio de la comparación de cadenas de ADN se logran conocer las similitudes entre ellas y estudiar las secciones desconocidas a partir de una base ya establecida. Tal que, a lo largo de esta investigación se pretende determinar si es posible optimizar los métodos de comparación de secuencias en cuanto a tiempo de ejecución y consumo de memoria por medio de la aplicación del concepto y tecnologías SIMD para la obtención de mejoras en el área de la Bioinformática.

II. Justificación

Hace cinco mil años, un hombre murió en los Alpes. Posiblemente, murió de un golpe en la cabeza, o desangrado hasta la muerte después de recibir un disparo de flecha en su hombro. Hay una gran cantidad de aspectos que no se conocen sobre este hombre, sin importar que los científicos lo han estado estudiando durante los últimos 30 años. Pero por otro lado, hay muchos aspectos que si se conocen, como el hecho de que el hombre poseía ojos cafés y una predisposición a enfermedades cardiovasculares, su tipo de sangre era O+. La capa que llevaba estaba conformada de parches de cuero de múltiples ovejas y cabras; y su sombrero hecho de piel de oso pardo. Toda esta información provino del ADN del hombre y de las prendas que usaba.

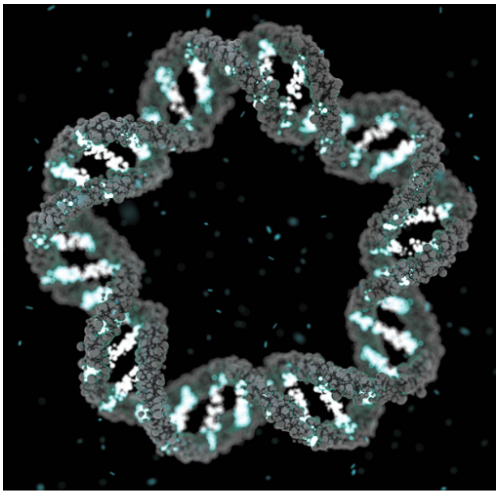


Illustration: Anatomy Blue

La molécula de ADN es una escalera en forma de doble hélice, conformada por miles de millones de bloques moleculares, cuya disposición determina muchos de los aspectos que nos hacen únicos a cada uno de nosotros. El ADN a su vez, puede almacenar grandes cantidades de información genética durante miles de años.

Recientemente, el ser humano ha contemplado el uso del ADN como un medio de almacenamiento de datos electrónicos y digitales, de forma tal que este podría ser una de las mejores alternativas para hacer frente a las grandes cantidades de datos, cada vez mayores, que se generan día con día y el costo de almacenamiento que esto representa. Tal es el caso, del CERN (European Organization for Nuclear Research), donde el enorme colisionador de hadrones genera 50 millones de GB de datos por año.

Con este objetivo en mente, es que resulta necesario la elaboración e investigación de nuevas tecnologías y procesos, tanto de hardware como de software, que vengan a revolucionar los diversos campos de estudio y las herramientas involucradas en el estudio de los organismos, su composición y comprensión del código genético, de forma que se generen factores de cambio y de avance en el área de la biotecnología que nos acerquen cada vez más a esta meta, aunque un poco utópica, de utilizar nuestro código genético, como un medio de almacenan miento de mayor capacidad que los existentes. Es aquí, donde la elaboración de esta investigación tiene una gran relevancia, porque si bien no nos vamos a centran en los procesos de almacenamiento, ni en una compresión a fondo de la composición del ADN; nos orientaremos al desarrollo y optimización de las herramientas básicas que se utilizan en el estudio de los fenómenos que dan inicio al estudio de estas áreas.

Siendo, de una gran importancia la agilización de procesos, como lo son los alineamientos y comprensión de las secuencias de ADN, para el estudio de los organismos conocidos y por conocer, y el reconocimiento de sus similitudes y capacidades. Así mismo, de una forma más general, esta investigación no solo llegaría a brindar conocimiento y ayuda en el campo de la biología, sino que podría ser implementada en el estudio de textos y patrones de escritura tanto actuales como ancestrales, brindado más herramientas en general al campo de estudio de secuencias de texto.

Finalmente, los datos recolectados de esta investigación estarán orientados a brindar nuevos puntos de vista a aquellas personas que se desarrollen en el campo de la bioinformática y que requieran de un alto nivel de procesamiento, tanto en cantidad de datos como en tiempo de ejecución, de forma que la utilización de las posibles optimizaciones de esta investigación, signifiquen una reducción de costos en cuanto al alquiler y/o compra de dispositivos de almacenamiento y/o de equipos de procesamiento para datos. Tal que, en un país como el nuestro, donde actualmente se están desarrollando diferentes avances en el campo de la biotecnología, se le puedan brindar herramientas que agilicen los procesos de investigación, dando mayor presencia al país e institución en el campo internacional

III. Marco Teórico / Estado del Arte

Las células de la mayoría de los organismos funcionan en formas similares. Tal que las proteínas producidas dentro de la célula de diferentes especies son muy similares entre sí, debido a que realizan las mismas tareas, como mantener los niveles de energía necesarios para que la célula funcione correctamente. Las proteínas por su parte, también tienen la misma tarea en diferentes organismos, como la detección de daños y la reparación del ADN, entre otros.

A demás, en el mundo de la biología constantemente se están buscando y analizando nuevos especímenes y muestras de organismos, muchos de ellos desconocidos; tal que, a partir de estos se obtienen nuevas cadenas de ADN a partir de la secuenciación del genoma, donde el primer paso a seguir con estas nuevas secuencias consiste en buscar las similitudes con secuencias conocidas en otros organismos. Si la función/estructura de una secuencias/proteínas se conocen, entonces es muy probable que la nueva secuencia corresponda a una proteína con la misma función/estructura. Tal que, a partir de la comparación de secuencias se puede obtener información relevante, como que en años atrás se encontró que solo el 1 % de los genes humanos no tiene una contra parte en el genoma del ratón y que la similitud media entre el ratón y los genes humanos son del 85 %. Tales similitudes existen porque todas las células poseen un ancestro en común.

Es por esto que la búsqueda de similitudes en las bases de datos de proteínas y ADN se ha convertido en una rutina esencial en los procedimientos de la Biología Molecular. Los algoritmos de Alineamiento Global y el algoritmo de Smith-Waterman (local), ha estado disponibles por más de 25 años. Se basan en un enfoque de programación dinámica que explora todas las posibles alineaciones entre dos secuencias; como resultado devuelven la alineación global o local óptima. Desafortunadamente, el costo computacional es muy alto, requiriendo una cantidad de operaciones proporcionales al producto de la longitud de dos secuencias. Además, del crecimiento exponencial de proteínas y bases de datos de ADN. Esto ha causado que la alineación progresiva que plantean estos algoritmos consuma mucho más tiempo y memoria. Tal que, la computación paralela podría ser una solución adecuada para tales aplicaciones.

III-A. Métodos de Alineamiento

EL objetivo general de la alineación de secuencias por pares se basa en encontrar la mejor combinación de dos secuencias, de forma que haya máxima correspondencia entre los residuos. Para conseguir este objetivo, una secuencia debe ser desplazada en relación con la otra para encontrar la posición donde se encuentran las máximas coincidencias. Para esto hay dos estrategias de alineación que se utilizan a menudo: alineación global y alineación local.

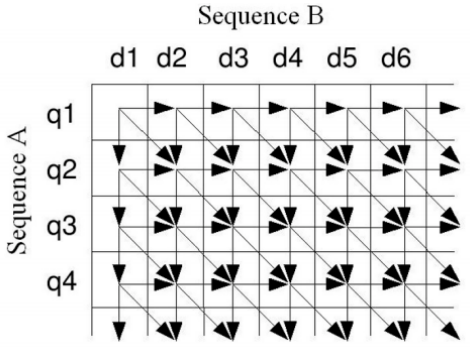
III-A1. Alineamiento Global: En el alineamiento global, las dos secuencias a ser alineadas generalmente se asumen similares sobre la totalidad de su longitud. Este método tiene la particularidad de que se realiza de inicio a fin de ambas secuencias para de esta manera encontrar la mejor alineación posible en toda la longitud entre las dos secuencias. Basados en el hecho de que ambas secuencias son sumamente similares, este método se aplica mayormente en la alineación de estrechamente relacionadas, de aproximadamente la misma longitud. Para secuencias divergentes y secuencias de longitud variables, este método no es capaz de generar resultados óptimos porque falla a la hora de reconocer regiones locales altamente similares entre las dos secuencias.

III-A2. Alineamiento Local: La alineación Local, por otro lado, no supone que las dos secuencias en cuestión tengan la misma longitud. Este método se enfoca en encontrar regiones locales con el nivel más alto de similitud entre las dos secuencias y alinea estas regiones sin tener en cuenta la alineación del resto de las regiones en la secuencias. Este enfoque puede ser para alinear secuencias más divergentes con el objetivo de buscar conservar los patrones en las secuencias de ADN o proteínas. Las dos secuencias a alinear pueden ser de diferentes longitudes. Este enfoque es más apropiado para alinear secuencias que contiene módulos similares.

III-B. SIMD

En computación, SIMD (Single Instruction, Multiple Data) es una técnica empleada para conseguir paralelismo a nivel de datos. Los repertorios SIMD consisten en instrucciones que aplican una misma operación sobre un conjunto de datos. Es una organización en donde una única unidad de control común despacha las

instrucciones a diferentes unidades de procesamiento. Todas éstas reciben la misma instrucción, pero operan sobre diferentes conjuntos de datos. Es decir, la misma instrucción es ejecutada de manera sincronizada por todas las unidades de procesamiento.



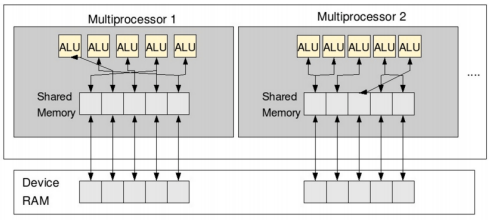
Smith-Waterman data dependencies. Each cell of the alignment matrix depends on the cells on the left and above it. Independent data can be found only on the same anti-diagonal.

III-B1. Extensiones SIMD: Corresponden a extensiones de ISA's existentes (ARMv7, x86, etc) para soportar paralelismo a nivel de datos. Creadas para el manejo de operaciones multimedia (audio, imágenes, gráficos). Normalmente, se tiene un procesador de propósito general, con hardware específico que permite la ejecución de instrucciones ensamblador para el manejo de operaciones vectoriales. La ventajas que presenta la utilización de estas radica en:

- Bajo costo en modificación de unidades aritméticas
- No requiere un alto ancho de banda.
- Usa transferencia de datos separada, para un grupo de operandos alineados en memoria (no hay problemas con falta de página en memoria virtual).
- Menos problemas con memoria cache.

III-B2. GPU's: Estas son unidades de procesamiento a nivel de hardware especializados en la generación de gráficos 2D y 3D, imágenes y vídeo, para sistemas operativos gráficos, vídeo juegos, sistemas de visualización (simulaciones), etc. Normalmente, utilizadas en aplicaciones con un alto nivel de procesamiento que requieren un gran nivel de paralelismo. Típicamente, son sistemas multinúcleo (arreglos paralelo de multiples procesadores gráficos), inmersos en sistemas heterogéneos (CPU + GPU).

Los dos principales proveedores de GPU, NVidia y AMD, poseen las plataformas de desarrollo, respectivamente, CUDA y CTM. A diferencia de los modelos de programación de GPU anteriores, estos son enfoques patentados diseñados para permitir un acceso directo a su hardware de gráficos específico. CUDA es una extensión del lenguaje de programación C; CTM es una máquina virtual que ejecuta código ensamblador propietario.



CUDA architecture. New CUDA compatible GPUs are implemented as a set of multiprocessors. Each multiprocessor has several ALUs (Arithmetic Logic Unit) that, at any given clock cycle, execute the same instructions but on different data. Each ALU can access (read and write) the multiprocessor *shared memory* and the device RAM.

III-C. CUDA

CUDA es una plataforma de computación paralela y un modelo de programación diseñado para NVidia para computación general en unidades de procesamiento gráfico (GPU). En las aplicaciones aceleradas por GPU, la parte secuencial de la carga de trabajo se ejecuta en la CPU, mientras que la parte de uso intensivo de cómputo se ejecuta en miles de núcleos de GPU en paralelo. Las GPU están organizadas en multiprocesadores, que agrupan múltiples procesadores de transmisión, las unidades de ejecución básicas. CUDA ejecuta el mismo programa en todos los multiprocesadores, el código para el programa es el mismo pero tanto los datos como el flujo de ejecución pueden ser diferentes y divergir. CUDA lanza múltiples instancias del mismo núcleo, llamadas hilos. Los hilos se agrupan en bloques para su ejecución. Los subprocesos son instancias de tiempo de ejecución del mismo kernel que ejecutan el mismo código; además, todos los subprocesos en un bloque son ejecutados por un multiprocesador de manera que ejecutan la misma instrucción al mismo tiempo, con diferentes datos.

IV. Objetivos

IV-A. General

- Determinar la mejor versión en cuanto a tiempo de ejecución y consumo de memoria de cada uno de los algoritmos de alineamiento.

IV-B. Específicos

- Evaluar la viabilidad de las extensiones tipo SIMD en la solución de problemas básicos de la Bioinformática .
- Evaluar la viabilidad del uso de GPU's en la solución de problemas básicos de la Bioinformática .
- Implementar soluciones de software paralelizable que hagan uso eficiente de los recursos del sistema.

V. Metodología

LA metodología a seguir para el desarrollo y obtención de datos de esta investigación se basara en un metodología experimental para la cual partiremos desde el punto de selección de multiples cadenas de ADN de la base de datos de AmtDB, el procesamiento de estas cadenas para generar sub cadenas modificadas y de diferentes longitudes, se procederá a desarrollar los códigos de procesamiento de las secuencias en el lenguaje de programación de C en cada una de sus versiones (secuencial, extensión SIMD, CUDA); seguidamente, se procederá a desarrollar el Benchmark especializado para la tabulación y medición de los tiempos de ejecución y niveles de memoria utilizados en cada uno de los casos de prueba. Cada uno de estos pasos se explicaran con más detalle a continuación, de forma que se brinde una idea clara del proceso a seguir y no quede duda alguna en el proceso de recolección, procesamiento de datos y reporte de resultas; tal que, los datos presentados en el paper final de la investigación resulten ser de la mayor confianza para el lector y tengas las menores fuentes de error posibles debido a la intersección humana durante el proceso, por lo cual se implementara un proceso de automatización completa para cada uno de los casos de estudio, este también se explicara a continuación.

V-A. Recolección y preparación de datos

Para lo que involucra el proceso de recolección de las cadenas de ADN se utilizara la base de datos AmtDB para la sección manual de los archivos que contiene las secuencias, ya que la base de datos de AmtDB provee un método de selección gráfica por medio de la ubicación de los ejemplares al rededor del globo, se seleccionaran un conjunto de 15 archivos poseedores de secuencias de ADN de un largo aproximado de 16 000 caracteres cada uno, de forma que cada uno de los 15 archivos sean seleccionados de diferentes locaciones geográficas y que no provengan del mismo país. Cada uno de estos archivos poseerá la extensión *.fa, correspondiente a la extensión de FASTA, formato de fichero informático basado en texto, utilizado para representar secuencias de ácidos nucleicos, péptidos, y en el que los pares de bases o los aminoácidos se representan usando códigos de una única letra. Estos archivos serán obtenidos durante tiempo de ejecución de la base de datos por medio de su dirección URL para ser descargados en el momento de su utilización, de forma que no se utiliza memoria de almacenamiento previo a su ejecución, sino solo durante su procesamiento, por lo tanto, la única referencia a estas será un archivo de texto (*.txt) con las URL de las 15 secuencias, una por linea en el archivo.

Estos archivos serán utilizados tanto en las versiones secuenciales de los dos algoritmos, como en las versiones SIMD. Para el caso del algoritmo de alineamiento Global (GSA), ya que con este se desea visualizar la similitud de forma completa entre la totalidad de los caracteres de ambas secuencias a comparar, no se realizara ningún procedimiento de preparación de secuencias previo a su uso en las versiones el método global. Más no es el mismo caso para el método local (LSA), debido a que este busca mantener los patrones de una forma más local entre dos secuencias de diferente longitud, para identificar posibles secuencias de proteínas ya conocidas en secuencias poco estudiadas, se procederá por medio del método de "shot gun", método del área de la biotecnología en el cual se "dispara"(corta) a una secuencia de ADN, reduciéndola a segmentos de menor longitud, para posteriormente, eliminar de forma aleatoria algunos de estos segmentos y unir los restantes. Tal que, de esta forma podremos aplicar el método Local para la comparación de la secuencia reducida contra su original y alinearla para

de esta forma identificar las secciones donde coinciden, e incluso dando inicio a un posible proceso de reconstrucción de genes, campo en el que también trabaja la bioinformática.

Ahora, que ya se obtuvieron las secuencias según los criterios mencionados anteriormente y mencionado el proceso de preparación previa para cada una delas secuencias dependiendo del algoritmo en el cual serán utilizadas, podemos proceder a la explicación de la metodología a seguir para la implementación de los algoritmo de alineamiento y cada una de sus versiones.

V-B. Desarrollo de los algoritmos

Como se menciono anteriormente, para esta investigación se trabajara sobre la implementación y optimización de dos algoritmos de alineamientos de secuencias (GSA y LSA) por medio de la utilización de herramientas, bibliotecas y hardware de tipo SIMD, utilizando la paralelización como método de optimización de procesamiento para el ahorro de tiempo, y mejor uso de la memoria por medio de buenas practicas de programación a la hora de ubicar datos en memoria, tomando en cuanta la forma en que el lenguaje de C ubica sus datos y estructuras en memoria.

Para cada uno de los algoritmos seguiremos la siguiente metodología:

1 - Investigar la teoría matemática detrás del algoritmo.

2 - Establecer el conjunto de ecuaciones matemáticas que definen los valores dentro de la matriz de puntajes a ser generada por cada algoritmo.

3 - Elaborar a partir de las ecuaciones previas los algoritmos necesarios para la implementación secuencial del algoritmo en cuestión.

4 - Realización de pruebas con secuencias pequeñas de ejemplo, para confirmar el funcionamiento del algoritmo.

5 - Basado en la teoría estudiada y las ecuaciones establecidas, deducir un nueva teoría de procesamiento de secuencias, sin modificar la logica detrás de cada método, para su paralelización, tomando en cuenta las herramientas SIMD. Considerando independencia de datos, matrices y procesos.

6 - Elaborar un conjunto de ecuaciones o pseudocódigo que brinden un modelo de programación para la implementación de las paralelizaciones.

7 - Elaborar a partir del modelo obtenido en el punto anterior la implementación del código de alineamiento en su segunda versión con extensiones SIMD (SSE).

8 - Realización de pruebas con secuencias pequeñas de ejemplo, para confirmar el funcionamiento del algoritmo en su nueva versión.

9 - Elaborar a partir del modelo obtenido en el punto 6, la implementación del código de alineamiento en su tercera versión para GPU's con CUDA.

10 - Realización de pruebas con secuencias pequeñas de ejemplo, para confirmar el funcionamiento del algoritmo en su nueva versión.

Los puntos descritos anteriormente, describen el procedimiento general a seguir para la construcción de cada uno de los algoritmos y sus versiones, junto con esta implementación se incluyen los procedimientos generales para la reconstrucción del alineamiento a partir de la matriz de puntajes y sub-rutinas asociadas a los procesos de inicialización de matrices, descarga de archivos, pre-procesamiento de secuencias, entre otros. En esta etapa de la implementación ya se tiene en cuenta e implementan métodos y buenas practicas de programación en cuanto al uso de memoria en algoritmos que involucran programación dinámica.

V-C. Desarrollo del Benchmark

En cuanto al desarrollo del sistema de evaluación de cada uno de los algoritmos en lo que respecta a su optimización en tiempo y mejora en el uso de memoria, se implementara un programa de evaluación el cual, tomara el archivo de texto con las URL's a cada una de las 15 secuencias, tomara cada una de las versiones de los algoritmos y los ejecutara dando como entrada de estos, combinaciones pre establecidas de las secuecnias de pruebas. Este tomara los tiempos de ejecución justo

antes de iniciar el proceso de alineamiento y justo después de que este termine, para seguidamente realizar la resta entre el $tiempo_final - tiempo_inicial = tiempo_total$. Así mismo, se medirá el uso de memoria al iniciar la ejecucíon del algoritmo, al finalizar este y en momento considerados críticos en cuanto a la cantidad de datos que pondrían existir en determinado momento para cada algoritmo, esta medición intermedia de la memoria será realizada evaluando cada uno de los procesos y seleccionando un punto en común entre las diferentes versiones de los algoritmos de alineamientos, para así, mantener un estándar de medición y reducir las fuentes de error que se puedan producir por la mala toma de datos entre las diferentes pruebas. De forma que este benchmark se implementará de forma que sea lo más determinista y consistente posible, sin importar las entradas y algoritmos que se encuentren bajo ejecución en un momento especifico.

V-D. Automatización

Una vez concluidas las etapas de recolección y preparación, construcción de los algoritmos y elaboración del benchmark, se procederá a la automatización completa del software de evaluación que dará los resultados finales y concluyentes sobre la ejecución y eficiencia de cada una de las versiones. De forma que la metodología de implementación de esta sección se basara en la separación de funcionalidades en común para la creación de una biblioteca dinámica que pueda ser utilizada entre los diferentes archivos, la utilización de Makefiles o CMake para los procesos de compilación de cada una de las versiones y su ejecución, así como una serie de comandos por consola para la configuración de las pruebas y corridas de los algoritmos.

Primeramente, se procederá a realizar la creación de la biblioteca de utilidades básicas compartidas entre los algoritmos y sus versiones. Seguidamente, y de ser necesario la construcción de una biblioteca que brinde las utilidades de medición y ejecución del benchmark o bien de los algoritmos y sus versiones implementadas. Para finalmente, concluir esta sección con la elaboración de los archivos de compilación de cada uno de los recursos necesarios, de forma que el proceso de realización de pruebas y obtención de mediciones sea un proceso sencillo y fácil de replicar para cualquier otra persona que desea utilizar el código aquí desarrollado para la reproducción de los resultados.

V-E. Presentación de Resultados

Los resultados a ser obtenidos por la ejecución de cada una de la versiones de los algoritmos (tiempo, uso de memoria, alineamiento obtenido y valor máximo de alineamiento) sean almacenados y representados de una forma accesible y legible para los lectores. Tal que, estos se almacenaran al final de la ejecución en archivo de texto plano, con un formato por definir (posiblemente JSON); de forma que sean tanto de fácil lectura para el usuario, como de sencillo procesamiento si se desea que este funcione de input para algún otro software lector de datos que este o pueda ser elaborado para la presentación gráfica de estos valores.

VI. Plan de Acción

Objetivo	Productos	Acciones	Semana
Obj. 1	Algoritmos con extensión SIMD	GSA con SIMD	16 y 17
Obj. 1		LSA con SIMD	16 y 17
Obj. 1	Obtención de Tiempo y uso de Memoria	Medición de Tiempo	16 y 17
Obj. 1		Medición de Memoria	16 y 17
Obj. 2	Algoritmos con CUDA	GSA con CUDA	16 y 17
Obj. 2		LSA con CUDA	16 y 17
Obj. 2	Obtención de Tiempo y uso de Memoria	Medición de Tiempo	16 y 17
Obj. 2		Medición de Memoria	16 y 17
Obj. 3	Automatización	Creación de Makefiles o CMake	16 y 17
Obj. 3	Uso correcto de Memoria	Medición total de Memoria	16 y 17

VII. Definición del cronograma

Actividad	Tiempo Estimado	Responsable
Selección de 15 Secuencias	1:00 h	Esteban
Preparación de Secuencias para el GSA	1:00 h	Esteban
Preparación de Secuencias para el LSA	2:00 h	Esteban
Implementa ción Secuencial del GSA	4:00 h	Esteban
Implementa ción Secuencial del LSA	4:00 h	Esteban
Implementa ción SIMD del GSA	8:00 h	Esteban
Implementa ción SIMD del LSA	8:00 h	Esteban
Implementa ción en CUDA del GSA	8:00 h	Esteban
Implementa ción en CUDA del LSA	8:00 h	Esteban
Desarrollo del Benchmark (Tiempos)	5:00 h	Esteban
Desarrollo del Benchmark (Memoria)	5:00 h	Esteban
Automatiza ción	8:00 h	Esteban
Presentación de Resultados	2:00 h	Esteban
Pruebas	8:00 h	Esteban
Evaluación de Resultados, Documentación interna y Paper con Resultados	12:00 h	Esteban
Total de Horas	$\sim$ 84:00 h	$\sim$ 3.5 días

VIII. Divulgación y transferencia de tecnología

Seguidamente, una vez concluida la investigación procesados, discutidos y generado un reporte de los resultados obtenidos a lo largo de esta investigación, en conjunto con el Paper final del proyecto podrían ser compartidos a la comunidad del Tecnológico de Costa Rica (TEC) por medio de la biblioteca o bien por medio de la Editorial del TEC en alguna de sus revistas sobre computación o bioinformática. Así mismo, podría compartirse a nivel nacional por medio de la búsqueda de alguna de las instituciones publicas en el área investigativa, menciónese el CENAT.

O bien, una de las propuestas a la hora de hablar sobre un proyecto en el área de la bioinformática con el profesor del curso electivo, de la Escuela de Computación del TEC, de Biología Molecular por Computadora, fue la opción de presentar el paper y los resultados de la presente investigación en el "Latin America High Performance Computing Conference (CARLA)", evento a llevarse a cabo el presente año en el mes de septiembre en el país, en el sector de Turrialba; evento que reúne a investigadores y estudiosos del área de la computación de alto rendimiento, para brindar exposiciones y charlas en las áreas de Computadores, Machine Learning, Bioinformática, entre otros temas.

IX. Presupuesto

Para el desarrollo de este proyecto se tiene que parte de las herramientas y recursos a ser utilizadas son de uso libre o de acceso publico, a excepción de los recursos físicos como el sistema de computo para desarrollar el código y realizar las pruebas secuenciales y de extensiones SIMD; y la tarjeta o tarjetas GPU's para la prueba de este tipo de versiones de los algoritmos. Tal que a continuación se mencionan los recursos a utilizar y su costo aproximado.

Recuso	Valor
AmtDataBase	free
CLion IDE License	$200
HP 240 G3 (8GB RAM / i3-4th / 500GB HDD)	$390
Jetson TX2 - GPU	$570
Total	$1,160

REFERENCIAS

1 O. Milenkovic, R. Gabrys, H. Kiah and S. Tabatabaei Yazdi, ."Exabytes in a Test Tube: The Case for DNA Data Storage", IEEE Spectrum, pp. 1-9, 2018.

2 J. Xiong, Essential Bioinformatics, 1st ed. Cambridge: Cambridge University Press, 2009, pp. 31-49 63-74.

3 S. Hochreiter, Bioinformatics I Sequence Analysis and Phylogenetics, 1st ed. A-4040 Linz, Austria: Institute of Bioinformatics Johannes Kepler University Linz, 2013, pp. 45-84.

4 S. Pawar, A. Stanam and Y. Zhu, ."Evaluating the computing efficiencies (specificity and sensitivity) of graphics processing unit (GPU)-accelerated DNA sequence alignment tools against central processing unit (CPU) alignment tool", Journal of Bioinformatics and Sequence Analysis, vol. 9, no. 2, pp. 1-5, 2018. Available: https://academicjournals.org/journal/JBSA/article-full-text-pdf/B39CEF858061. [Accessed 23 May 2019].

5 P. Patil, P. Pattiwar, S. Khan and V. Panch, "Speed-up of Sequence Alignment Algorithms on CUDA Compatible GPUs", International Journal of Research in Engineering, Science and Management, vol. 1, no. 12, pp. 1-5, 2018. Available: https://www.ijresm.com/Vol_1_2018/Vol1_Iss1_December18/IJRESM_V1_I12_145.pdf. [Accessed 23 May 2019].

6 C. Hung a, Y. Lin, C. Lin, Y. Chung and Y. Chung, ÇUDA ClustalW: An efficient parallel algorithm for progressive multiple sequence alignment on Multi-GPUs", ELSEVIER, vol. 1, no. 3, pp. 1-7, 2015. Available: http://cs.nthu.edu.tw/ ychung/journal/computational-biology-chimistry-2015.pdf. [Accessed 23 May 2019].

7 S. Manavski and G. Valle, ÇUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment", BMC Bioinformatics, vol. 9, no. 2, pp. 1-9, 2008. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-S2-S10. [Accessed 23 May 2019].

8 N. Homer, B. Merriman and S. Nelson, "Local alignment of two-base encoded DNA sequence", BMC Bioinformatics, vol. 10, no. 1, pp. 1-11, 2009. Available: https://bmcbioinformatics.biomedcentral.com/track/pdf/10.1186/1471-2105-10-175. [Accessed 23 May 2019].

9 CUDA C PROGRAMMING GUIDE, 10th ed. Santa Clara, CA 95050: NVIDIA Corporation, 2019.