PROTOTIPO DE ARQUITECTURA DE COMPUTACIÓN RECONFIGURABLE PARA EL ALGORITMO DE BLASTN

Prototype of Reconfigurable Computing architecture for algorithm BLASTN

RESUMEN
Blast es uno de los programas más utilizados por los investigadores en Bioinformática [1]. Este software es en realidad una recopilación de cinco subprogramas (blastn, blastp, blastx, tblastn y tblastx). Para su ejecución en un proyecto real, requiere de sistemas de cómputo con altas capacidades de procesamiento y memoria. Al reemplazar la función de búsqueda que se ejecutaba en los procesadores, por una que se procese sobre una plataforma de computación reconfigurable, permite liberar al procesador de dicha tarea y puede significar una reducción en tiempos de ejecución, proporcionando gran ayuda a las investigaciones en secuenciación genética.

PALABRAS CLAVES: Blast, Bioinformática, computación reconfigurable, secuenciación genética.

ABSTRACT
Blast is one of the most used by Bioninformatics researchers. This software is actually a compilation of five sub-programs (blastn, blastp, blastx, tblastn and tblastx). For its implementation in a real project, requires computer systems with high processing capabilities and memory. By replacing the search function that was running on a processor to be processed on a reconfigurable computing platform, freeing up the processor for that task and may mean a reduction in execution time, providing great assistance in the investigation into gene sequencing
KEYWORDS: *Bioinformatics, Blast, Reconfigurable Computing, gene sequencing.*

RAMIRO ANDRÉS BARRIOS VALENCIA
Ingeniero de Sistemas y Computación
Profesor Auxiliar
Universidad Tecnológica de Pereira
ramiro@sirius.utp.edu.co

JOSE ALFREDO JARAMILLO VILLEGAS
Ingeniero Electrónico.
Profesor Asistente
Universidad Tecnológica de Pereira
jj@sirius.utp.edu.co

JUAN DAVID HINCAPIÉ ZEA
Ingeniero Electrónico.
Profesor Auxiliar
Universidad Tecnológica de Pereira
judaz@sirius.utp.edu.co

1. INTRODUCCIÓN

Desde que BLAST es una herramienta Open Source, científicos y biólogos han hecho de ésta su herramienta favorita para investigaciones de diferentes propósitos como son: determinación de filogenia, secuenciación genética, estudios patológicos en epidemiologia, entre otros. Diferentes grupos, tanto en la industria como en la academia han tratado de migrar el algoritmo BLAST a diferentes tipos de arquitecturas paralelas como los son Grids, Clusters, Asics y otras de diferentes maneras [2] [3] [4] [5] [6].
Muchas de éstas implementaciones proporcionan una salida diferente al original BLAST [7] perdiendo todo contacto con la versión original, impidiéndole a ésta realizar todos los cálculos necesario para mostrar la mayoría de datos estadísticos y de resumen. Por otro lado, el hecho de tener que migrar las herramientas completas, de un grupo que desea utilizar dicha implementación, lo hace complejo y poco práctico. Por ellos se podrían plantear las siguientes preguntas:

- En investigaciones y/o desarrollos que utilizan BLAST, ¿puede mejorar la producción científica una aceleración de dicha herramientas?
- De las herramientas y esfuerzos presentes en el mercado o la sociedad científica, ¿son accesibles estos a grupos de investigación o científicos interesados en su uso?
- ¿Son estas herramientas utilizables por estos grupos en sus investigaciones teniendo en cuenta que muchas de ellas no entregan los mismos reportes que el original BLAST?

La principal motivación al realizar este proyecto, fue la de mejorar y ofrecer una solución propia al problema de los altos tiempos de ejecución de BLAST, y proporcionar de ésta manera un mayor aprovechamiento de los tiempos en la ejecución de proyectos que requieren su uso. Como consecuencia directa del anterior punto, se pretende entonces incrementar la producción en investigaciones relacionadas.

2. ANÁLISIS DE SECUENCIAS: ALINEAMIENTO Y COMPARACIÓN DE SECUENCIAS

En la Tabla 1 se enumeran los tipos de análisis más frecuentes a los que se someten las secuencias biológicas, su correspondiente descripción y herramientas utilizadas para ello. De ésta, se describirán más detalladamente las

Fecha de Recepción: 25 de Enero de 2011
Fecha de Aceptación: 11 de Abril de 2011

actividades de alineamiento de secuencias y búsquedas de secuencias en bases de datos, ya que es importante tener claro el proceso y su importancia para el entendimiento del presente proyecto.

La forma más común de comparar dos secuencias es realizar alineamientos de secuencias**,** lo que da información sobre las regiones semejantes entre ambas y mejor aun es comparar una secuencia con las bases de datos ya existentes, ya que los programas que realizan esta tarea se basan en alineamientos de una secuencia con todas (una por una) de la base de datos.

El objetivo que se persigue al realizar un alineamiento entre dos secuencias es determinar si poseen suficiente similitud como para poder justificar la existencia de homología entre ellas. Aunque ambos términos, similitud y homología, a menudo se confunden y se usan indistintamente, existiendo una clara diferencia entre los dos.

- ✓ La similitud es un concepto cuantificable, que puede medirse y expresarse como un porcentaje de identidad entre dos secuencias.
- ✓ La homología se refiere a una conclusión obtenida de la similitud, e indica si dos secuencias están, o no, relacionadas o comparten una historia evolutiva común.

Que se hace	Para qué se hace	Herramientas
Encontrar Genes	Identificar posibles regiones codificadoras en las secuencias genómicas de ADN	GENSCAN, GensWise, PROCUSTES, GRAIL
Detección de características de ADN	Localizar sitios de unión, promotores y secuencias relacionadas con la regulación en la expresión de Genes	CBS Prediction Server
Traslación y vuelta a tras de traslación en ADN	Convertir una secuencia de ADN en proteína o viceversa	"Protein Machina" servidor en EBI
Alineamiento de Pares de secuencias (Local)	Localizar regiones cortas de homología entre un par de secuencias largas	BLAST y FASTA
Alineamiento de Pares de secuencias (Global)	Bucar el mejor alineamiento "a todo lo largo" entre dos secuencias	ALIGN
Buscador de secuencias en Base de datos para comparación entre pares	Busca concordancia de secuencias que no son reconocidas por buscadores de palabras clave: encuentra solo concordancia que en verdad tienen alguna homología entre las secuencias.	BLAST, FASTA y SSEARCH

Tabla 1. Resumen de análisis de secuencias y herramientas.

2.1. Comparación por identidad

Su algoritmia consiste en desplazar una secuencia debajo de la otra anotando el número de coincidencias que ocurren, seleccionando como resultado la posición de mayor valor (Ver Figura 2).

Se ilustran seis iteraciones del algoritmo (Figura 1), correspondientes a: las tres primeras, la de mayor puntuación y las dos últimas. Después de realizar todas las iteraciones necesarias, de cada una de ellas se conservan las duplas que contienen la puntuación (número de símbolos que coincidieron) y el identificador de la iteración (tabla llamada T, ver Figura 2). Estos son posteriormente comparados para obtener el mayor de ellos y arrojar el resultado (tal como se muestra en la iteración 9 de la Figura 1) gracias a la información salvada en la tabla llamada I de la Figura 2.

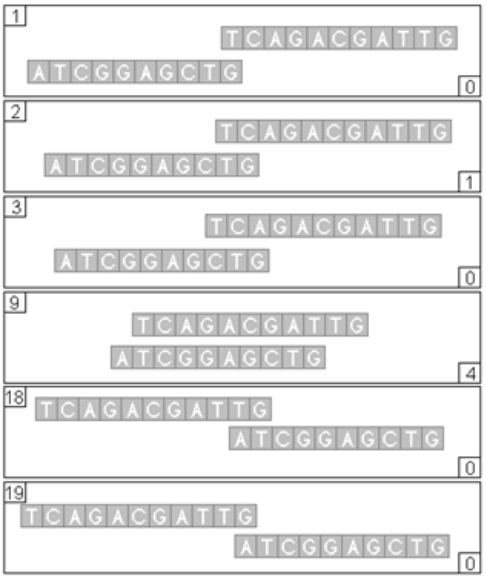

Figura 1. Iteraciones del algoritmo de comparación por identidad.

2.1. Comparación por semejanza: DotPlot

El procedimiento de éste tipo de comparación se puede visualizar mediante una matriz (Figura 3). De esta manera se observa la diagonal correspondiente a cada una de las iteraciones en el subproceso de alineamiento y se puede determinar cuál de ellas es la de mayor puntaje tal como aparece en la figura.

T

Iteracion	Punt.
1	0
2	1
3	0
4	2
5	0
6	0
7	3
8	1
9	4
10	3
11	2
12	3
13	2
14	2
15	0
16	1
17	3
18	1
19	0
20	0

I

ID	Pos(S_1)	Pos(S_2)
1	1	10
2	1	9
3	1	8
4	1	7
5	1	6
6	1	5
7	1	4
8	1	3
9	1	2
10	1	1
11	2	1
12	3	1
13	4	1
14	5	1
15	6	1
16	7	1
17	8	1
18	9	1
19	10	1
20	11	1

Figura 2. Descripción cuantitativa de las iteraciones del algoritmo de comparación por identidad

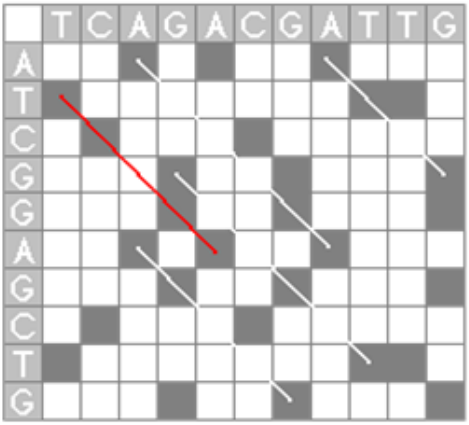

Figura 3. Algoritmo DotPlot para el mismo ejemplo de del apartado anterior.

3. BLAST

Los cinco programas tradicionales de BLAST son: BLASTN, BLASTP, BLASTX, TBLASTN y TBLASTX [8]. En Tabla 2 se puede observar cada uno de ellos con sus correspondientes usos y que tipo de bases de datos y secuencias *query* utiliza.

Programa	Bases de datos	Consulta (query)	Usos típicos
BLASTN	Nucleótidos	Nucleótidos	Mapeo de oligonucleótidos, cADN, y productos PCR a un genoma; exploracion de secuencias a través de
BLASTP	Proteínas	Proteínas	Identificar regiones comunes entre proteínas; colección de proteínas relacionadas por análisis
BLASTX	Proteínas	Nucleótidos trasladados	Encontrar gnes que codifican en proteínas en ADN genómico; determinar si el cADN corresponde a una
TBLASTN	Nucleótidos trasladados	Proteínas	Identificacion de transcripts, potencialmente desde múltiples organismos, similares a una proteína
TBLASTX	Nucleótidos trasladados	Nucleótidos trasladados	Prediccion de genes a través de especies en los genomas o a nivel de transcripts; búsqueda de genes

Tabla 2. Descripción de los sub-programas de BLAST

3.1. El Algoritmo de BLAST

El espacio de búsqueda entre dos secuencias puede ser visualizado como un grafo con una secuencia a lo largo de eje X y la otra a lo largo del eje Y. Cada emparejamiento en este espacio es representado mediante un punto tal como se puede ver en Figura 4. Cada par de letras emparejadas tiene un puntaje determinado mediante la matriz de sustitución o mediante el esquema de puntuación seleccionado. Un alineamiento es una secuencia de letras emparejadas que pueden contener gaps. Los alineamientos sin gaps aparecen como líneas diagonales en el espacio de búsqueda y su puntaje es la suma de los emparejamientos (match o mismatch) menos el costo total de sus gaps[1].

Para reducir el espacio de búsqueda BLAST [8] utiliza tres capas de reglas para refinar secuencialmente emparejamiento de alto puntaje (de ahora en adelante HSP, las siglas de inglés de High Scoring Pairs). Estas capas heurísticas conocidas como seeding (ensemillado), extention (extensión) y evaluation (evaluación), son procedimientos de refinamiento gradual que permite a BLAST reducir el espacio de búsqueda para no malgastar tiempo en regiones disimilares.

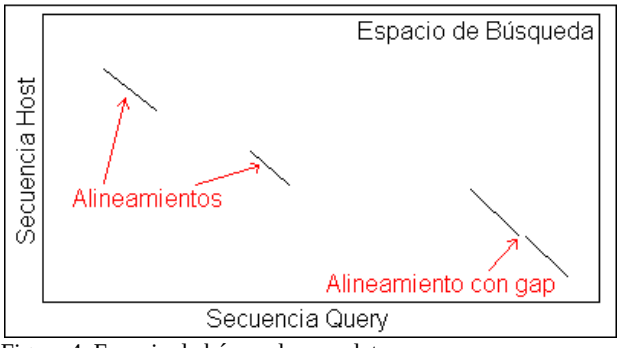

Figura 4. Espacio de búsqueda completo.

BLAST asume que alineamientos significativos tienen palabras en común. Una palabra es un número definido de letras. Por ejemplo, si se define una palabra de tres letras, la secuencias ATGGCA, tendría las palabras ATG,

[1] El termino gaps es acuñado a "huecos" o espacios no significativos en la secuencia nucleótica o de proteínas.

TGG, GGC y GCA. Al comparar dos secuencias, BLAST primero determina la posición de todas las palabras comunes que son llamadas palabras éxito (Hits). Esto puede ser visto en la Figura 5 y usando solo las regiones con palabras éxito (Hits), se puede ignorar un gran espacio de búsqueda. Esto se conoce como proceso de sembrado.

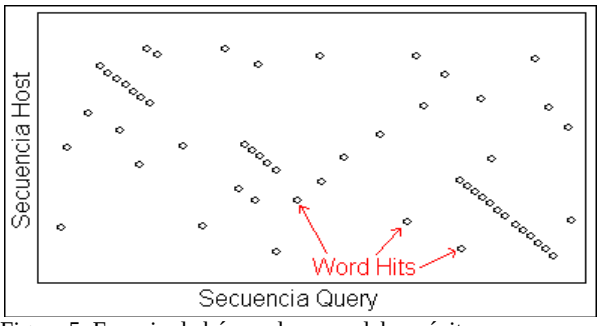

Figura 5. Espacio de búsqueda con palabras éxito

Una vez el espacio de búsqueda ha sido sembrado, los alineamientos pueden ser generados desde las semillas individuales. La extensión puede ser vista gráficamente como unas flechas que se dirigen en sentidos contrarios desde una palabra éxito como se muestra en la Figura 6.

Una vez las semillas son extendidas en ambas direcciones para crear los alineamientos, éstos son evaluados parar determinar si son estadísticamente significantes. Esos que son estadísticamente significativos son llamados HSPs. En un nivel simple, evaluarlos sería fácil, se necesitaría un umbral de puntaje, que se llamará S, para organizar los alineamientos entre los más bajos y más altos puntajes. Debido a que S y E están directamente relacionados con la ecuación de Karlin-Altschul [9], el umbral de puntaje es sinónimo al umbral estadístico. Pero, en la práctica evaluar alineamientos no es tan simple debido a la complicación que surge de tener varios HSPs.

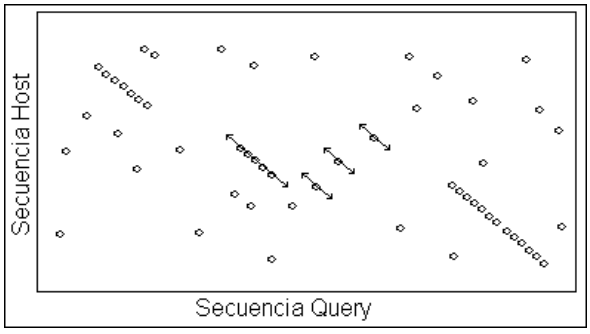

Figura 6. Proceso de extensión.

En BLAST, el parámetro –e, en la línea de comandos, pone el valor para el umbral final. El valor para el umbral de alineamiento es puesto por el software y no puede ser modificado por el usuario.

4. DISEÑO PROPUESTO

El diseño propuesto reemplaza la función encargada de encontrar los hits del software de BLAST compilado de NCBI[2], que es el Centro Nacional de Información Biotecnológica de Estados Unidos de sus siglas en inglés. Esta lista es posteriormente pasada a la función encargada de realizar la extensión y la evaluación de los éxitos encontrados. Esto permitió que la ejecución de BLAST en la arquitectura de computación reconfigurable sea transparente al usuario.

En la Figura 7 se muestra el diagrama de bloques que describe de manera general el flujo de procesos dentro de la Unidad Lógica, llamada BPU. Éste es el diseño a implementar en la FPGA de RASC, que es un sistema de computación reconfigurable de SGI disponible en el Supercomputador SGI Altix 350 de la Universidad Tecnológica de Pereira. Esta implementación se hará sobre una Virtex II 6000 implantada dentro del modulo de RASC, utilizando el lenguaje de descripción de hardware VHDL.

En el bloque de procesamiento denominado BPU (Por las siglas en inglés de BLAST Proccessing Unit), está encargado de encontrar los emparejamientos (matches units) y posteriormente la determinación de éxitos (hits units).

Y para finalizar a partir de los éxitos encontrados se realiza el cálculo de la posiciones en query y subject (query es la secuencia consultadas y subject la secuencia objetivo en la base de datos) correspondientes al éxito, datos que son almacenados en memoria para ser pasados de retorno al software

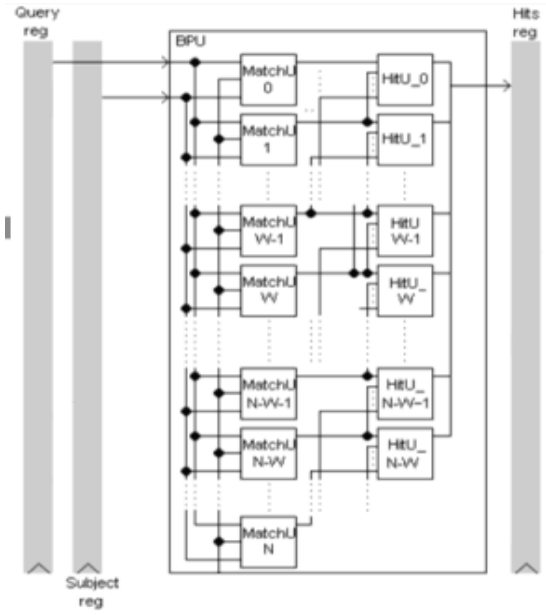

Figura 7.Diagrama de Bloques del diseño propuesto.

[2] NCBI. National Center for Biotechnology Information. Ver http://www.ncbi.nlm.nih.gov/

5. RESULTADOS OBTENIDOS

Debido a la simplicidad del diseño en hardware, la cantidad de unidades de *matches units* y *hits units* que pueden ser sintetizadas en BPU permitieron analizar secuencias de hasta 2000 nucleótidos en un solo ciclo de reloj de 100 Mhz.

Las pruebas de Software Vs Computación Reconfigurable, alcanzó una aceleración de hasta 10X del algoritmo en computación reconfigurable con respecto al de software debido a las características de concurrencias en la implementación de la FPGA. La limitante de lograr mayores aceleraciones se atribuye a la alta tasa de transferencia y el ancho de banda limitado disponible en la arquitectura.

6. BIBLIOGRAFÍA

[1]. Korf, Ian, Yandell, Mark y Bedell, Joseph. BLAST. Sebastopol, CA. United States of America : O' reilly, 2003. págs. 75-95. Caps. 5 y 12.

[2]. Bjornson, R. D., y otros. TurboBLAST: A Parallel Implementation of BLAST Built on the TurboHub. TurboGenomics, Inc. April 2003. Proceedings of the Third IEEE International Workshop on High Performance Computational Biology.

[3]. Camp, Nick, Cofer, Haruna y Gomperts, Roberto. High throughput - blast. SGI. 1998.

[4]. RC-BLAST: towards a portable, cost-effective open source hardware implementation. Muriki, K., Underwood, K. D. y Sass, R. April 2005, Proceedings. 19th IEEE International Parallel and Distributed Processing Symposium, 2005. , pág. 8 pp. ISBN: 0-7695-2312-9.

[5]. Blast++: A tool for blasting queries in batches. Ong, Twee Hee, y otros. Abril, 2003. In Proceedings of the Third IEEE International Workshop on High Performance Computational Biology.

[6]. A General Reconfigurable Architecture for the BLAST Algorithm. Sotiriades, Euripides y Dollas, Apostolos. 3, s.l. : Springer Netherlands, Junio, 2007, The Journal of VLSI Signal Processing, Vol. 48, págs. 189-208. ISBN: 0922-5773 .

[7]. ftp de NCBI para BLAST. descargas de fuentes, ejecutables bases de datos y documentacion sobre BLAST. [En línea] NCBI. [Citado el: 20 de 06 de 2008.] ftp://ftp.ncbi.nlm.nih.gov/blast/.

[8]. BLAST. [En línea] NCBI. [Citado el: 20 de 06 de 2008.] http://blast.ncbi.nlm.nih.gov/Blast.cgi

[9]. Karlin, S. & Altschul, S.F. (1990) "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes." Proc. Natl. Acad. Sci. USA 87:2264-2268.