DISEÑO DE ACELERADORES PARA EL ALINEAMIENTO GLOBAL DE SECUENCIAS DE ADN

Martin A. Lozano, Vladimir Trujillo, Jaime Velasco-Medina

Grupo de Bio-nanoelectrónica
EIEE, Universidad del Valle, A.A. 25360, Cali, Colombia
E-mail: martloza, vlatruo, jvelasco@univalle.edu.co

ABSTRACT

Este artículo presenta el diseño de dos aceleradores hardware para el alineamiento global de secuencias de ADN. El algoritmo implementado es el algoritmo de Needleman y Wunsch, el cual es basado en usar programación dinámica. Las arquitecturas son basadas en un arreglo 2-D y un arreglo sistólico, los cuales permiten calcular los valores de la matriz de puntuación, el principal proceso del algoritmo. En este caso, los arreglos 2-D y sistólico realizan el cálculo de los valores ponderados de la matriz usando procesamiento paralelo y secuencial respectivamente. Los diseños se realizaron usando captura esquemática y descripción estructural en VHDL, la síntesis y simulación se realizó usando Quartus II versión 5 de Altera, y los diseños fueron sintetizados en la Stratix II EP2S130F1020C4 (arreglo 2-D) y EP2S15F484C3 (arreglo sistólico). Los resultados de las simulaciones muestran que las dos implementaciones hardware producen un mejor alineamiento global que los presentados por los programas software Muscle v3.6 y MB advanced DNA análisis versión 6.82.

1. INTRODUCCION

Debido al proyecto del Genoma Humano en la actualidad existe una gran cantidad de información sobre la composición de los aminoácidos, por lo tanto se debe recurrir al uso de herramientas bioinformáticas que permitan procesar rápidamente esta información.

En este contexto, el alineamiento global de dos secuencias de ADN permite conocer la relación evolutiva, es decir conocer si las dos secuencias de ADN tienen un ancestro común del cual pudieron haber evolucionado por bifurcación de especies o por duplicación de genes, sin embargo el alineamiento global de secuencias de ADN se debe formalizar rigurosamente desde el punto de vista matemático. En este caso, el alineamiento global óptimo entre dos secuencias de ADN es aquel que hace máxima la suma de las puntuaciones de los residuos alineados de acuerdo con una matriz de substitución dada. En principio, es posible determinar el alineamiento global óptimo entre dos secuencias elaborando todos los alineamientos posibles y calculando para cada uno de ellos la suma de las puntuaciones de los residuos alineados. Sin embargo, el problema es muy difícil de resolver cuando existen "gaps", en este caso el número de alineamientos posibles entre las dos secuencias de ADN es muy grande y el cálculo exhaustivo es muy difícil de alcanzar usando los actuales recursos computacionales [1].

Determinar el grado de semejanza entre dos secuencias de ADN requiere elaborar el alineamiento global y contar (directa o indirectamente) el número de posiciones equivalentes conservadas. Cuando la relación entre dos secuencias es lejana y exhiben una pobre conservación global elaborar el alineamiento no es trivial, puesto que es difícil (si no imposible) identificar los aminoácidos equivalentes en las dos secuencias, en particular porque residuos individuales e incluso regiones enteras de la proteína pueden haber desaparecido (o haberse insertado) a lo largo de la evolución en una de las dos secuencias. Entonces, varios algoritmos son utilizados e implementados en software, los cuales son simples, utilizan pocos recursos computacionales, emplean poco tiempo y los resultados presentan un nivel de acierto muy aceptable. Sin embargo, la desventaja es que los resultados generados son de índole heurístico, es decir, los resultados son aproximados, debido a que los alineamientos son realizados por métodos de probabilidad.

Este trabajo presenta dos implementaciones en hardware del algoritmo *Needleman-Wunsch* para el Alineamiento Global de Secuencias de ADN, el cual presenta mejores resultados que las implementaciones en software tales como: mayor velocidad de procesamiento y resultados exactos.

El trabajo está organizado de la siguiente forma: En la sección 2 se describen los trabajos previos, en la sección 3 se describen los algoritmos básicos para el alineamiento de secuencias de ADN, en la sección 4 se presenta el diseño de los aceleradores, en la sección 5 se presentan los resultados de simulación y finalmente, en la sección 6 se presentan las conclusiones y el trabajo futuro.

2. TRABAJOS PREVIOS

En la literatura revisada se encuentran pocas publicaciones sobre la implementación en hardware del algoritmo *Needleman-Wunsch* para alineamiento global

de secuencias de ADN y adicionalmente estas publicaciones no presentan o describen en forma explicita los resultados de desempeño obtenidos. B. Wang en [2] presenta la implementación de un procesador basado en una arquitectura Cell Matrix usando programación dinámica. En este caso, el trabajo propuesto solo presenta la implementación del proceso del cálculo de la matriz de puntuación y el proceso de la recuperación de las secuencias alineadas no es considerado. Por lo tanto, en [2] no se implementa completamente el algoritmo *Needleman-Wunsch* y los resultados de simulación presentados son para secuencias de 11 bases de longitud.

T. V. Court y M. C. Herbort en [3] presentan de forma muy general una arquitectura hardware que permite realizar el alineamiento global de dos secuencia de ADN. Sin embargo en este trabajo los resultados no son bien descriptos y sustentados desde el punto de vista de diseño de procesadores o hardware digital.

3. ALGORITMOS DE ALINEAMIENTO DE SECUENCIAS DE ADN

Existe una gran variedad de algoritmos que pueden ser usados para evaluar similitudes en secuencias de proteínas y ADN [4-9]. En este caso, la selección del mejor algoritmo depende del problema a resolver. Dos algoritmos óptimos para realizar el alineamiento de secuencias de ADN han sido presentados en [10] y [11].

En [10], se presenta el algoritmo de *Needleman-Wunsch* (1970), el cual calcula los puntajes de similitud global entre dos secuencias. En [11], se presenta el algoritmo de *Smith-Waterman* (1981), el cual calcula los puntajes de similitud local.

Los algoritmos que realizan las comparaciones locales (encontrar similitudes entre dos secuencias, ignorando diferencias fuera de las regiones más parecidas) son los más apropiados para hacer búsquedas de proteínas y ADN en bases de datos, mientras los algoritmos que realizan la comparación global son los más apropiados para construir árboles evolutivos, es decir cuando las homologías se han establecido previamente [1].

4. DISEÑO DE LAS ARQUITECTURAS DE LOS ACELERADORES DE ADN

Las arquitecturas de los aceleradores diseñados implementan en forma exacta el algoritmo propuesto por *Needleman y Wunsch* para el alineamiento global de secuencias de ADN..

En este trabajo, el cálculo de los valores ponderados de la matriz de puntuación se realiza por medio de un arreglo 2-D o un arreglo sistólico.

4.1. Arreglo 2-D

En la Figura 1, se muestra la arquitectura para reusar una matriz de puntuación básica de n x n, en este caso, se usan registros de desplazamiento, los cuales almacenan las secuencias a alinear. Cada vez que la matriz de puntuación realiza el cálculo de los valores de puntuación, estos se almacenan en una memoria y los valores de la última fila y la última columna de la matriz de puntuación se almacenan en un registro. Después de realizar el cálculo de la matriz de puntuación, se hace un desplazamiento en los registros que contienen las secuencias a alinear y mediante el uso de multiplexores se seleccionan los nuevos valores para calcular los siguientes valores de puntuación.

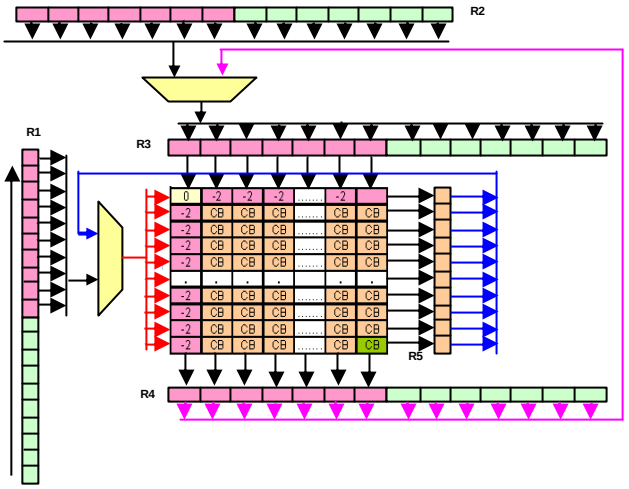

Figura 1. Arreglo 2-D de NxM basado en un arreglo de n x n

4.2 Arreglo sistólico

En la Figura 2 se muestra como calcular la matriz de puntuación usando un arreglo sistólico, el cual procesa en forma secuencial las dos secuencias. En este caso, se usa un vector que realiza un recorrido en forma diagonal dentro de la matriz. La ventaja de este arreglo es que permite calcular los valores de la matriz de puntuación usando un solo vector, el cual tiene un tamaño mucho menor que el arreglo 2D. Los datos entrada para el vector son los valores ponderados vecinos a la posición actual del vector y los valores de las secuencias a comparar, reduciendo ampliamente el área utilizada para calcular los valores de la matriz de puntuación.

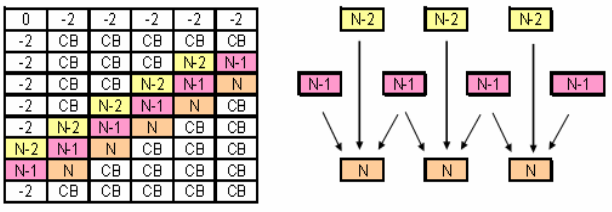

Figura 2. Arreglo sistólico: vectores N-2, N-1 y N

4.3 Arquitecturas de los aceleradores de ADN

En la Figura 3 se muestra el diagrama de bloques de la arquitectura de un acelerador de ADN. Los bloques funcionales son: registros de entrada y salida, la unidad de cálculo de la matriz de puntuacion la cual puede ser a arreglo 2-D o un arreglo sistolico, la unidad de

"traceback", la unidad de inserción de "gaps" y la unidad de control la cual es la encargada de controlar el flujo de los datos. Para ambos diseños, el alineamiento de las secuencias se realiza a partir de la información que se calcula y almacena en la matriz de puntuación. En el caso del arreglo 2D, cuando se obtienen todos los valores de la matriz de puntuación también se obtienen las secuencias alineadas por la unidad el "traceback", a partir de esta información la unidad de inserción de "gaps" genera el alineamiento optimo de las dos secuencias

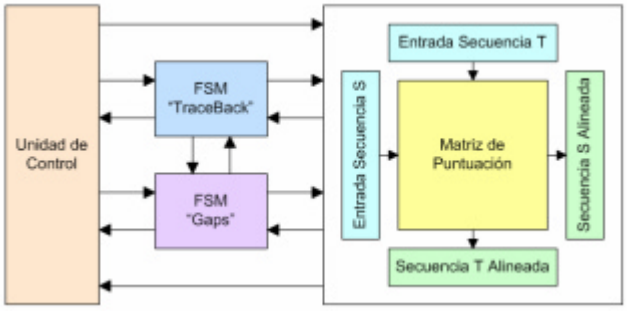

Figura 3. Arquitectura del acelerador de ADN

El bloque funcional que realiza el cálculo de los valores ponderados de la matriz de puntuación es la celda básica, la cual es implementa usando circuitos combinatorios.

5. RESULTADOS DE SIMULACIÓN

Con el propósito de verificar el funcionamiento de los aceleradores de ADN, varias simulaciones fueron llevadas a cabo. En las Figuras 4, 5 y 6 se muestran los resultados de simulación para alinear dos secuencias, T (25 bases) y S (16 bases). La Figura 4 muestra los resultados de simulación de la arquitectura hardware usando un arreglo 2-D, la Figura 5 muestra los resultados generados por el programa Muscle v3.6 [12] y la Figura 6 muestra los resultados entregados por el programa MB Advanced DNA Análisis versión 6.82 [13]. En este caso, las secuencias para alinear son:

T(25) = TTTTTGAACCAACCGCAAGGTTCCA
S(16) = TTAAAACCCGAAGGCA

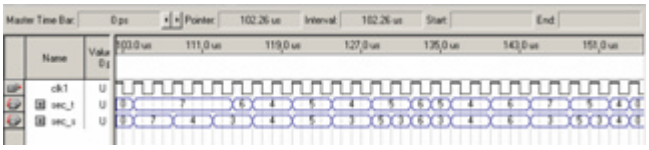

Figura 4. Resultados de simulación para el alineamiento global de las secuencias T(25) y S(16)

Desde los resultados de simulación de la Figura 4, se puede observar que el *ancestro común* esta compuesto por 14 bases: T T A A C C C G A A G G C A . En este caso, el alineamiento óptimo de las dos secuencias es:

T(25) = TTTTTGAACCAACCGCAAGGTTCCA
S(16) = TTAA--AACC - - C- G -AAGG - - C -A

```
seq1          tttTTgAaccAACCgcAAGGttcCA
seq2          ---TTaA---AACCcgAAGG---CA
```

Figura 5. Resultado de simulación usando Muscle v3.6 para las secuencias T(25) y S(16)

Desde los resultados de simulación de la Figura 5, se puede observar que el *ancestro común* esta compuesto por 13 bases: T T A A A C C A A G G C A. En este caso, el alineamiento óptimo de las dos secuencias es:

T(25) = TTTTTGAACCAACCGCAAGGTTCCA
S(16) = - - - TTAA - - - AACCCGAAGG- - - CA

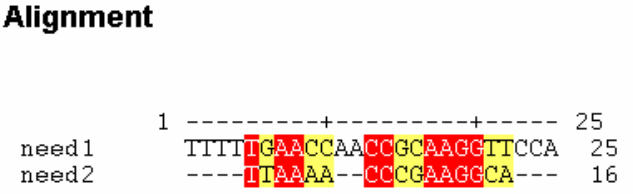

Figura 6. Resultado de simulación usando MB Advanced DNA Análisis para secuencias T(25) y S(16)

Desde los resultados de simulación de la Figura 6, se puede observar que el *ancestro común* esta compuesto por 9 bases: T A A C C A A G G. En este caso, el alineamiento óptimo de las dos secuencias es:

T(25) = TTTTTGAACCAACCGCAAGGTTCCA
S(16) = - - - - TTAA AA- - CCCGAAGGCA - - -

También, el módulo bioinformática del programa Matlab ha sido utilizado para alinear las dos anteriores secuencias. La Figura 7 muestra los resultados de simulación generados por el programa Matlab.

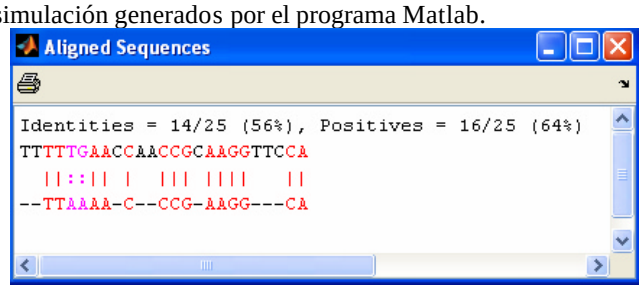

Figura 7. Resultados de simulación usando Matlab para las secuencias T(25) y S(16)

Desde los resultados de simulación de la Figura 7, se puede observar que el ancestro común esta compuesto por 14 bases: T T A A C C C G A A G G C A. En este caso, el alineamiento óptimo de las dos secuencias es:

T(25) = TTTTTGAACCAACCGCAAGGTTCCA
S(16) = - - TTAAAA- C - - CCG -AAGG - - - CA

En las Figuras 8, 9 y 10 se muestran los resultados de simulación para alinear dos secuencias, T (30 bases) y S (19 bases). La Figura 8 muestra los resultados de simulación de la arquitectura hardware usando un arreglo sistólico, la Figura 9 muestra los resultados generados por el programa Muscle v3.6 y la Figura 10 muestra los resultados entregados por el programa MB Advanced DNA Análisis versión 6.82. En este caso, las secuencias para alinear son:

T(30) = TTCTGTTTTTGAACCAACCGCAAGGTTCCA
S(19) = ACGTTAAAACCCGAAGGCA

Figura 8. Resultados de simulación para el alineamiento global de las secuencias T(30) y S(19)

Desde los resultados de simulación de la Figura 8, se puede observar que el *ancestro común* esta compuesto por 16 bases: C G T T A A C C C G A A G G C A. El alineamiento óptimo de las dos secuencias es:

T(30) = TTCTGTTTTTGAACCAACCGCAAGGTTCCA

S(19) = A- C- GTTAA- - AACC - -C -G - AAGG- - C- A

```
seq1            ttctgtttTTgAaccAACCgcAAGGttcCA
seq2            -----acgTTaA---AACCcgAAGG---CA
```

Figura 9. Resultado de simulación usando Muscle v3.6 para las secuencias T(30) y S(19)

Desde los resultados de simulación de la Figura 9, se puede observar que el *ancestro común* esta compuesto por 13 bases: T T A A A C C A A G G C A . En este caso, el alineamiento óptimo de las dos secuencias es:

T(30) = TTCTGTTTTTGAACCAACCGCAAGGTTCCA

S(19) = - - - - --ACG TTAA - - - AACCCGAAGG - - - CA

Alignment

```
           1 ---------+---------+---------+--- 33
sec1         TTCTGTTTTTGAAC-C-AA-CCGCAAGGTTCCA  30
sec2         -ACGTTAAAACC--C-GA-A-GGCA--------  19
```

Figura 10. Resultado de simulación usando MB Advanced DNA Análisis para secuencias T(30) y S(19)

Desde los resultados de simulación de la Figura 10, se puede observar que el *ancestro común* esta compuesto por 6 bases: C T A G C A. En este caso, el alineamiento óptimo de las dos secuencias es:

T(30)= TTCTGTTTTTGAAC-C- AA- CCGCAAGGTTCCA

S(19)= - ACGTTAAAACC- -C-GA -A- GGCA - - - - - - - -

La Figura 11 muestra los resultados de simulación generados por el programa Matlab.

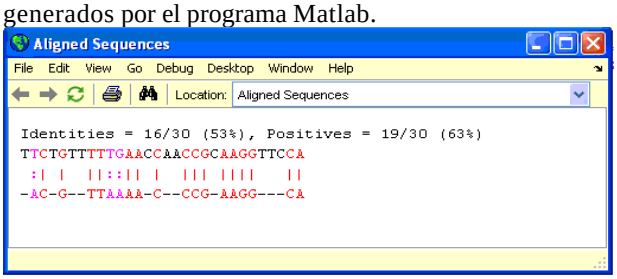

Figura 11. Resultados de simulación usando Matlab para las secuencias T(30) y S(19)

6. CONCLUSIONES Y TRABAJO FUTURO

Este artículo presenta el diseño de aceleradores hardware para el alineamiento global de secuencias de ADN. El algoritmo implementado es el algoritmo de *Needleman y Wunsch*, el cual es basado en usar programación dinámica.

Las arquitecturas implementadas son basadas en un arreglo 2-D y un arreglo sistólico, los cuales permiten calcular los valores de la matriz de puntuación, el principal proceso del algoritmo. En este caso, los arreglos 2-D y sistólico realizan el cálculo de los valores ponderados de la matriz. Los diseños se realizaron usando captura esquemática y descripción estructural en VHDL, la síntesis y simulación se realizó usando Quartus II versión 5 de Altera, y los diseños fueron sintetizados en la Stratix II EP2S130F1020C4 (arreglo 2-D) y EP2S15F484C3 (arreglo sistólico).

Los resultados de las simulaciones muestran que las dos implementaciones hardware producen un mejor alineamiento global que los presentados por los programas software Muscle v3.6 y MB advanced DNA análisis versión 6.82. En este contexto, desde los resultados de simulación se puede inferir que para alinear secuencias del orden de 500 bases en adelante, los aceleradores hardware pueden ser una solución eficiente.

El trabajo futuro será orientado inicialmente a diseñar un procesador que permita el alineamiento global de dos secuencias de ADN con una longitud de 500 bases. Posteriormente, el trabajo será orientado a diseñar procesadores genómicos y proteómicos.

REFERENCES

[1] Roderic Guigó I Serra, "Bioinformática: La creciente interconexión entre la biología y la informática", Boletín Electrónico de la Sociedad Española de Genética, pp. 4, Julio 2003.

[2] Bin Wang, "Implementation of a Dynamic Programming Algorithm for ADN Sequence Alignment on the Cell Matrix Architecture" http://www.cellmatrix.com/entryway/products/pub/wang2002.pdf

[3] Tom Van Court, "Families of FPGA-Based Accelerators for Approximate String Matching", www.bu.edu/caadlab/05Match Ext.pdf

[4] "Alineamiento de Secuencias: Programación Dinámica" http://www.ccpg.fq.edu.uy/Cursos/ BIOINF101/Slides/Clase_04 / Clase04_6.pdf

[5] Z. Luthey-Schulten, "Sequence and Structure Alignment", http://www.ks.uiuc.edu/Training/SumSchool/ 2004/materials/ lectures/Day6/Mon22a.pdf

[6] Marina Alexandersson, "Sequence Analysis – Pairwise Sequence alignment", http://www.fcc.chalmers.se/~marina/ files/BioI_PairAlign_2005.pdf.

[7] Oswaldo Trelles, "Comparación de Secuencias Biológicas Algoritmia", http://ub.cbm.uam.es/support/courses/ Leon2005_arrays/alignment/CompBioS-Alg.pdf

[8] Vladimir Likic, "The Needleman–Wunsch Algorithm for Sequence Alignment", http://www.ludwig.edu.au/course/ lectures2005/Likic.pdf

[9] David J. Lipman, Stephen F. Altschul, and John D. Kececioglu, "A tool for Multiple Sequence Alignment", http://www.pnas.org/ content /vol86/issue12.

[10] Saul B. Needleman, Christian D. Wunsch, "A General Method Applicable to Search for Similarities in the Amino Acid Sequence of Two Proteins", *J. Mol. Biol.*, 48, pp. 443-453, 1970, http://www.cs.umd.edu/class/spring2003/cmsc 838t/papers/needlemanandwunsch1970.pdf

[11] http://www.cecalc.ula.ve/bioinformatica/BIOTUTOR/ tutoriales.htm

[12] http://phylogenomics.berkeley.edu/

[13] http://www.softpile.com/Education/Science/Review_19837_ 19837_index.html