FARAUTE Ciens. y Tec., 2(2): 46-53, 2007 ISSN 1698-7418
Depósito Legal PP200402CA1617

CONSTRUCCIÓN DE UNA BASE DE DATOS GENÓMICA Y SU EMPLEO MEDIANTE UNA APLICACIÓN DE ANÁLISIS DE SECUENCIAS

Construction of a Genomic Database and Its Use by a Sequence Analysis Application

JORGE J. RANGEL-LAGARDERA[1], PEDRO LINARES H.[2]
Universidad de Carabobo. Facultad de Ciencia y Tecnología. [1]Departamento de Biología [2]Centro de Visualización y Cómputo Científico. Carabobo. Venezuela.
{jjrangel1, plinares}@uc.edu.ve

Fecha de Recepción: 15/02/2007, **Fecha de Revisión:** 25/09/2007, **Fecha de Aceptación:** 30/10/2007

Resumen

Se desarrolló una aplicación de software de análisis de secuencias biológicas. Este incluye funciones básicas relacionadas con el flujo de información genética, tales como: Replicación, secuencias invertidas, transcripción, transcripción inversa y traducción. La aplicación GCMP (Siglas en inglés de "Programa Manipulador del Código Genético") contiene una implementación del algoritmo de alineamiento global, Needleman-Wunsch, tanto para proteínas como para ADN. También se implementó el algoritmo Smith-Waterman para comparación de pares de secuencias o de una secuencia con una base de datos (BD) local. Dicha BD contiene el genoma de 21 microorganismos procariotas, además de los siguientes eucariotas: *Homo sapiens*, *Mus musculus*, *Drosophila melanogaster*, *Caenorhabditis elegans*, y *Pan troglodytes*. Los alineamientos locales encontrados en la BD pueden ser redirigidos directamente por el programa hacia un visualizador de genes en Internet. El método de búsqueda separa los genomas de la BD en cadenas del doble del tamaño de la secuencia a comparar, y además solapa partes de dos cadenas consecutivas con el propósito de realizar una comparación exhaustiva. GCMP consta de una interfaz gráfica amigable, escrito el lenguaje de programación Python.
Palabras claves: Alineamiento de secuencias, bioinformática, genómica.

Abstract

We have developed a software application for biological sequence analysis. The program includes basic functions related to the genetic information flow, such as: Replication, inverted sequences, transcription, reverse transcription and translation. GCMP (Genetic Code Manipulating Program) contains a global alignment implementation, the Needleman-Wunsch algorithm, for proteins as well as DNA. The Smith-Waterman algorithm was also implemented, which allows pairwise sequence comparison or a local database (DB) search. This DB includes genomes of 21 prokaryote microorganisms, as well as the following eukaryotes: *Homo sapiens*, *Mus musculus*, *Drosophila melanogaster*, *Caenorhabditis elegans*, and *Pan troglodytes*. Local alignments found in the DB can be redirected by the program to an internet gene viewer. The search method splits genomes in strings twice the size of the query sequence, and it also overlaps consecutive string fragments with the intention of performing an exhaustive comparison. GCMP possess a friendly graphic interface, it is written in the programming language Python.
Key words: Bioinformatics, genomics, sequence alignment

1. Introducción

La bioinformática representa un área nueva, creciente e interdisciplinaria de la ciencia que usa enfoques computacionales para responder preguntas biológicas (Mount, 2004). Esta disciplina tiene aplicaciones notorias en campos tan diversos como: sistemática, taxonomía, biología molecular, bioquímica, evolución, biomedicina, entre otros.

La secuenciación del primer genoma de un organismo de vida libre, el de la bacteria *Haemophilus influenzae* (Fleischmann *et al.*, 1995), marcó el advenimiento de la era genómica en biología. Posteriormente, más de 400 procariotas han sido objeto de la misma clase de proyectos, así como también varios eucariotas, entre los que se encuentran: La mosca de la fruta *Drosophila melanogaster* (Adams *et al*., 2000), el nemátodo *Caenorhabditis elegans* (*C. elegans* Sequencing Consortium, 1998), la levadura *Saccharomyces cerevisiae* (Cherry *et al*., 1997), la planta *Arabidopsis thaliana* (*Arabidopsis* Genome Initiative, 2000), el ser humano (Lander *et al.*, 2001; Venter *et al.*, 2001).

La gran cantidad de información genética recopilada en las bases de datos, proveniente de los proyectos de secuenciación y otras investigaciones, necesita ser analizada para llegar a conclusiones útiles. Con este fin, se han desarrollado una variedad de algoritmos para el análisis de secuencias (Needleman & Wunsch, 1970; Smith & Waterman, 1981; Gotoh, 1982; Altschul *et al.*, 1990). El alineamiento de secuencias es uno de los problemas más importantes de la bioinformática, ya que este procedimiento tiene numerosas aplicaciones, desde la identificación molecular, pasando por la dilucidación de las relaciones evolutivas de los organismos hasta el descubrimiento de la función de nuevos genes.

Existen dos clases de alineamiento. El primero es el alineamiento global, el cual consiste en comparar dos o más secuencias buscando patrones de caracteres que estén en el mismo orden y que además, incluyan la totalidad de éstas. En segundo lugar, los alineamientos locales son regiones de similitud pero, a diferencia del alineamiento global, consisten de subsecuencias que contienen la mayor densidad de coincidencias.

Aunque existen muchos programas disponibles en servidores en Internet para la búsqueda en bases de datos (Lipman & Pearson, 1985; Altschul *et al.*, 1990), estos sitios se congestionan como consecuencia del intenso uso por los usuarios.

Por ello, esta investigación se propuso desarrollar una aplicación de software capaz de realizar búsquedas de secuencias en una base de datos local. Además, el programa también permite realizar una multitud de tareas relacionadas con genética molecular.

Para mejorar la legibilidad y la eficiencia en el mantenimiento del código fuente, se escogió un lenguaje de programación con dichas prestaciones.

2. Materiales y Métodos

La aplicación (GCMP, siglas de "Genetic Code Manipulating Program") fue escrita en Python 2.4, cuyo intérprete emplea los paquetes wxPython 2.7 y NumPy 1.0. Este lenguaje de alto nivel le otorgó al programa portabilidad entre los sistemas operativos Linux y Windows.

Tipo de secuencia	Función
ADN	Obtener la cadena complementaria de ADN
	Obtener la cadena complementaria invertida
	Obtener la secuencia invertida
	Transcribir la secuencia a ARN
ARN	Obtener la cadena de ADN complementaria (cADN)
	Traducir la secuencia a proteína
Proteína	Generar posible secuencia de ARN codificadora aleatoriamente

Tabla 1. Funciones del programa relacionadas con el flujo de información genética para distintas secuencias.

La función de traducción de secuencias de ARN (Tabla 1) estuvo basada en el código genético universal, mientras que la simulación de los procesos de obtención de la cadena complementaria y transcripción se hicieron siguiendo las reglas de apareamiento de bases de Watson-Crick (Watson & Crick, 1953). La generación de una posible secuencia de ARN codificadora fue escrita con la ayuda de un módulo de la Biblioteca de Referencia de Python, la cual emplea el generador de números pseudoaleatorios Mersenne-Twister.

El programa contiene implementaciones de los algoritmos de programación dinámica Needleman-Wunsch (Needleman & Wunsch, 1970) y Smith-Waterman (Smith & Waterman, 1981) para alineamientos de pares de secuencias, así como búsqueda en bases de datos.

Las secuencias de ADN de la base de datos local se almacenaron en formato FASTA. Los genomas disponibles localmente pertenecen a 21 microorganismos procariotas de importancia científica (Tabla 2) y a los siguientes eucariotas: Ser humano (*Homo sapiens*, 3.000 Mb*)*, un nemátodo (*Caenorhabditis elegans*, 100 Mb*)*, el ratón (*Mus musculus*, 3.000 Mb*)*, la mosca de la fruta (*Drosophila melanogaster*, 122 Mb*)* y el chimpancé (*Pan troglodytes*, 3.000 Mb*)*.

También se implementó una función de búsqueda que reconoce secuencias para las enzimas de restricción *Eco*RI, *Bam*HI y *Hind*III.

2.1 Algoritmo de Needleman-Wunsch

Dadas las secuencias $a = a_1a_2\ldots a_m$ (secuencia de búsqueda, en el caso de búsqueda en bases de datos) y $b = b_1b_2\ldots b_n$ (secuencia de base de datos), dónde $n \geq m$, se crean dos matrices (*M* y *P*) de m+1 filas y n+1 columnas. Los recorridos dentro de una matriz representan todos los posibles alineamientos, permitiendo a cada letra emparejarse con una letra igual o distinta en la otra secuencia, así como también hacerla coincidir con un gap (representado con el símbolo '-').

Organismo	Tamaño del genoma (Mb)
Clostridium tetani	2,87
Corynebacterium glutamicum	3,3
Clamidia muridarum	1,08
Escherichia coli	4.64
Desulfovibrio vulgaris	3,7
Micoplasma pneumoniae	0,82
Nanoarchaeum equitans	0,49
Nitrobacter hamburgensis	5,01
Pseudomonas aeruginosa	6,3
Psychrobacter arcticus	2,65
Mycobacterium avium	5,5
Methanococcus maripaludis	1,66
Leptospira borgpetersenii	3,92
Haemophilus influenzae	1,83
Halobacterium sp.	2,57
Erythrobacter litorales	3,05
Caulobacter crescentus	4,02
Bdellovibrio bacteriovorus	3,78
Bacillus licheniformis	4,22
Bacillus thuringiensis	5,28
Bacillus subtilis	4,21

Tabla 2. Genomas de microorganismos procariotas disponibles en la base de datos local de GCMP.

El algoritmo Needleman-Wunsch (Needleman & Wunsch, 1970) encuentra el recorrido óptimo dentro de la matriz calculando valores para cada una de las celdas. La primera fila y la primera columna de la matriz se computan de acuerdo con las siguientes relaciones:

$$M(0,0) = 0$$
$$M(0,i) = -i * d \qquad (1)$$
$$M(j,0) = -j * d$$

El valor del resto de las celdas se calcula de la siguiente forma: dada una penalización *d* para la inserción de un gap, una penalización *s* para comparación de bases distintas y una puntuación t para comparación de bases iguales, el valor de la celda *M(i,j)* (Excepto para las celdas de la primera fila y la primera columna) se determina según las siguientes relaciones recursivas:

$$M(i,j) = \max \begin{cases} F(i-1,j-1) + s(a_i, b_j) \\ \text{(Bases iguales o distintas)} \\ F(i-1,j) - d \\ \text{(Introducción de gap en secuencia b)} \\ F(i,j-1) - d \\ \text{(Introducción de gap en secuencia a)} \end{cases} \qquad (2)$$

El máximo de las tres ecuaciones es escogido y colocado en la celda respectiva. Simultáneamente, el valor de la celda correspondiente de la matriz P es asignado dependiendo del máximo escogido para la matriz M. Si se decide emparejar dos bases iguales o distintas, el valor 2 será almacenado en la matriz, lo cual significa que el movimiento en M debe ser diagonal. En cambio, si el máximo correspondió a la asignación de un gap a la base de la secuencia b, ó a, se almacenará un tres, ó un uno, en la matriz P, respectivamente.

La construcción de las secuencias alineadas es llevada a cabo rastreando en reversa M, desde la parte inferior derecha hasta la superior izquierda, empleando como guía la matriz P, de acuerdo con las reglas descritas antes.

2.2 Algoritmo Smith-Waterman.

Este algoritmo constituye una modificación del método Needleman-Wunsch (Smith & Waterman, 1981), con el propósito de encontrar subsecuencias biológicas en lugar de alinear la totalidad de las secuencias (alineamiento global).

Los cambios al algoritmo Needleman-Wunsch consisten en rellenar la primera fila y la primera columna de M con ceros. Las otras celdas son tratadas tal como se explicó antes, con la excepción de que no se permiten números negativos. En caso de que el máximo de las tres ecuaciones sea un valor negativo, se almacenará un cero en su lugar. Esta ligera modificación sirve de guía para que el algoritmo detenga la construccion del alineamiento para puntuaciones demasiado bajas. Los alineamientos se empiezan a construir localizando la puntuación máxima en M. Nótese que es posible que en M existan varios valores de puntuaciones altas, y por lo tanto, varios alineamientos locales para un par de secuencias. Por ello, se estableció un valor límite (threshold), el cual es la puntuación mínima requerida para iniciar la construcción de un alineamiento.

Durante la búsqueda en la base de datos, el programa ejecuta la implementación de Smith-Waterman sucesivamente con la secuencia de búsqueda y fragmentos de secuencias de los genomas. En cada comparación, los genomas son separados en cadenas de 2*len(s) de longitud (Dónde len(s) es la longitud de la secuencia de búsqueda). Para asegurar que se encuentren secuencias similares, aún cuando la región homóloga se encuentre dividida en dos cadenas consecutivas, la siguiente cadena a examinar contiene parte de la anterior (en otras palabras, se solapan), específicamente len(s)/2 nucleótidos.

3. Resultados y discusión.

Una de las principales virtudes de GCMP es su interfaz gráfica amigable para el usuario (Fig. 1).

Las tareas de análisis de secuencias y flujo de información genética se encuentran en el menú “Operaciones”. Éstas se encuentran agrupadas de acuerdo con el tipo de secuencia sobre la que actúan (ADN, ARN o proteína).

Fig. 1. Apariencia general de GCMP

Algunas de las funciones del programa más empleadas, como búsqueda en bases de datos o abrir archivos de secuencias, se encuentran en una barra de herramientas. Sin embargo, las tareas relacionadas con el flujo de información solamente son accesibles a través de los menús. Las capacidades de alineamiento de secuencias de la aplicación también aparecen en el menú correspondiente al tipo de secuencia que se está analizando. En la Fig. 2 se observa una secuencia previamente cargada, justo antes de ser analizada, para obtener su cadena complementaria.

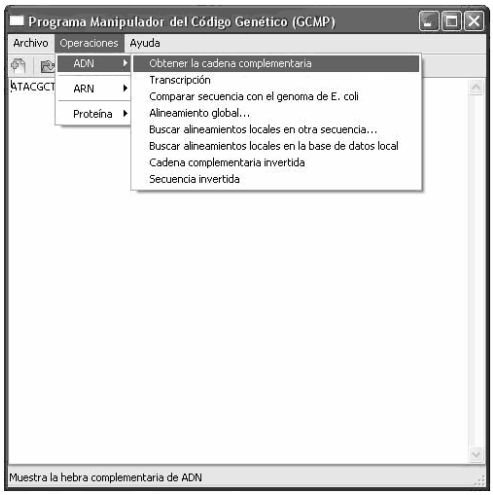

Fig. 2. Funcionamiento de la interfaz gráfica de GCMP

La Fig. 3 muestra el resultado de la obtención de la cadena complementaria.

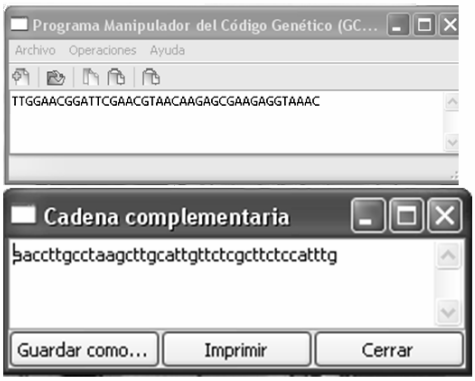

Fig. 3. Operación de obtener la cadena de ADN complementaria

Las operaciones de alineamientos globales requieren cargar las dos secuencias a comparar. El alineamiento óptimo se muestra en una ventana aparte (Fig. 4). Las dos secuencias se colocan una encima de otra, de modo que sus bases puedan compararse y las regiones homólogas se hagan visibles. Los gaps se representan con el símbolo '-', los cuales modelan las mutaciones ocurridas durante el proceso de divergencia evolutiva entre dos secuencias. Hay que destacar que el sistema de puntuación afecta la manera en que se insertan los gaps y se alinean los caracteres.

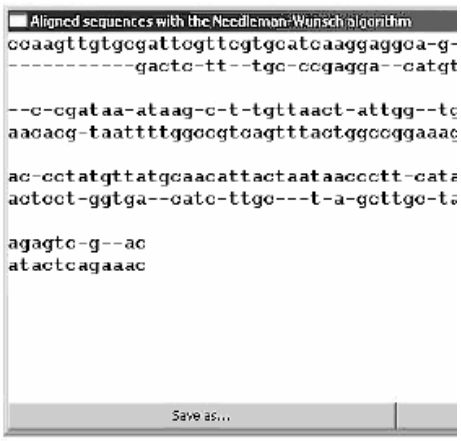

Fig. 4. Alineamiento global entre dos secuencias de ADN

La función de búsqueda en la base de datos es la que exige más poder de procesamiento. Una secuencia de búsqueda abierta con el programa es comparada con todas las secuencias de la base de datos mediante la implementación de Smith-Waterman. Aunque los métodos basados en k-tuples (Lipman & Pearson, 1985; Altschul *et al*., 1990), algoritmos que emplean pequeñas palabras construidas a partir de la secuencia de búsqueda para realizar comparaciones, son más rápidos que los enfoques de programación dinámica (Smith & Waterman, 1981), se ha reportado que éstas últimas metodologías detectan más eficazmente las secuencias lejanamente emparentadas (Anderson & Brass, 1998).

Un archivo HTML, con los alineamientos y la puntuación encontrados, es la salida de esta función del programa. Este documento contiene enlaces, para cada alineamiento, hacia la herramienta de visualización de genes NCBI Sequence Viewer. De esta manera, se puede observar directamente el gen encontrado, así como sus secuencias vecinas en el genoma.

Con el propósito de demostrar la función de búsqueda en la base de datos, se generó una secuencia aleatoria de 1000 nucleótidos y se

comparó con los genomas almacenados en el programa. Dos alineamientos locales encontrados por el programa se muestran en la Fig. 5.

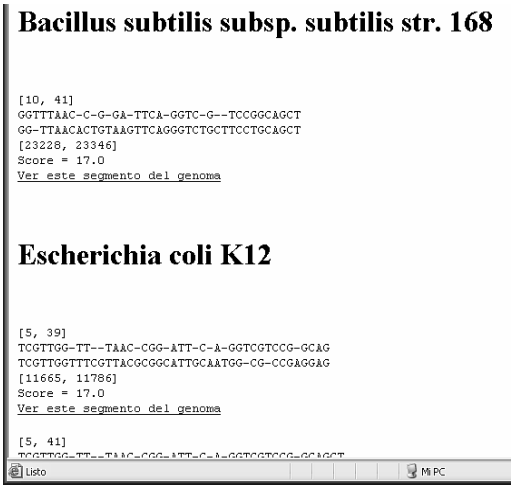

Fig. 5. Archivo HTML generado por GCMP mostrando secuencias encontradas en la base de datos

Nótese los enlaces con las palabras "Ver este segmento del genoma", que redirigen al visualizador de genes mencionado antes.

Todas las operaciones que GCMP contiene son ampliamente usadas en la biología molecular moderna. Además, la interfaz del programa es muy fácil de usar, por lo cual sus aplicaciones se extienden desde el campo de la investigación hasta el de la docencia. Aunque una búsqueda en toda la base de datos puede llevar algunos minutos, esta desventaja queda compensada con una mejora en los resultados obtenidos. Gracias a que los genomas se encuentran almacenados en el computador, la aplicación resuelve el problema de las colas en los servidores remotos.

Desde el punto de vista de la implementación, la legibilidad del código fuente de Python facilita el mantenimiento y las futuras mejoras. Adicionalmente, la Biblioteca de Referencia del lenguaje y la disponibilidad de paquetes como ByoPython permiten acelerar aún más la velocidad de desarrollo de nuevos algoritmos.

Una versión ampliada de este software será empleada en un futuro proyecto de investigación, cuyo objetivo será evaluar la biodiversidad microbiana del Centro Termal Las Trincheras. En el diseño experimental de dicho trabajo se recolectarán secuencias de microorganismos mediante la Reacción en Cadena de la Polimerasa (PCR) y secuenciación, para luego ser comparadas con la base de datos del programa. De esta manera, se identificarán algunos géneros y especies presentes en la comunidad.

4. Bibliografía

Adams, M., S. Holt S, E. Gocayne, A. Scherer, L. Hoskins & R. Galle. (2000). The genome sequence of *Drosophila melanogaster*. *Science.* 287:2185-2195.

Altschul, S., W. Gish, W. Miller, E. Myers & D. Lipman. (1990). Basic Local Alignment Search Tool. *J. Mol. Biol.* 215:403-410.

Anderson, I. & Brass, A. (1998). Searching DNA databases for similarities to DNA sequences: When is a match significant?. *Bioinformatics*. 14:349-356.

Arabidopsis Genome Initiative. (2000). Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature*. 408:796-815.

C. elegans Sequencing Consortium. (1998). Genome sequence of the nematode *C. elegans*: A platform for investigating biology. *Science*. 282:2012-2018.

Cherry, J.M., C. Ball, S. Weng, G. Juvik, R. Schidt, C. Adler, B. Dunn, S. Dwight, L. Riles, R. Mortimer & D. Bolstein. (1997). Genetic and physical maps of *Saccharomyces cerevisiae*. *Nature*. 387:67-73.

Fleischmann, R., D. Adams, O. White, R. Clayton, E. Kirkness, A. Kerlavage, C. Bult, J. Tomb, B. Dougherty & J. Merrick. (1995). Whole genome random sequencing and assembly of *Haemophilus influenzae. Science.* 269:496-512.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162:705-708.

Lander, E., L. Linton, B. Birren, C. Nusbaum, M. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, R. Funke & D. Gage. (2001). Initial sequencing and analysis of the human genome. *Nature*. 409:860-921.

Lipman, D. & W. Pearson. (1985). Rapid and sensitive protein similarity searches. *Science*. 227:1435-1441.

Mount, D. (2004). *Bioinformatics Sequence and Genome Analysis.* Cold Spring Harbor Laboratory Press. New York.

Needleman, S. & C. Wunsch. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol*. 48:443-453.

Smith, T. & M. Waterman. (1981). Identification of common molecular subsequences. *J. Mol. Biol.* 147:195-197.

Venter, J., M. Adams, E. Myers, P. Li, R. Mural, G. Sutton, H. Smith, M. Yandell, C. Evans, R. Holt & J. Gocayne. (2001). The sequence of the human genome. *Science*. 291:1304-1351.

Watson, J. & F. Crick. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*. 171:737-738.