Uso de Shingling para completar información de genes de la Pseudomona Aeruginosa AG1

Jorge Cedeño-Fernández
Maestría en Ciencias de la Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
E-mail: jcedeno@itcr.ac.cr

Francisco J. Torres-Rojas
Escuela de Ingeniería en Cumputación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
Email: torres@ic-itcr.ac.cr

***Resumen*—El algoritmo de Shingling fue propuesto por Andrei Z. Broder como una solución para establecer un grado de similitud de los documentos en la Web y facilitar la indexación de los mismos. Se ha evidenciado el potencial del algoritmo para agrupar conjuntos de genes similares. Algunos de los genes del conjunto total carecen de información que se puede inferir o asociar de otros genes que sí la poseen, esto considerando que el algoritmo de Shingling permite agruparlos por su similitud. Con un grado de similitud asociado a los genes se pueden enfocar esfuerzos de análisis con el uso de algoritmos más precisos como lo es el alineamiento global de secuencias Needleman-Wunsch. El presente paper abarca la aplicación del algoritmo de Shingling mejorado para el procesamiento, búsqueda, comparación y asociación de genes de acuerdo con su similitud.**

Palabras clave*—*Bioinformática, gen, shingling, pseudomona aeruginosa, AG1, ADN, ARN.

I. Introducción

La cepa AG1 de la bacteria Pseudomona Aeruginosa (PA) carece de un transcriptoma, que es una colección de todas las lecturas o transcritos de genes presentes en una célula [1], debido a la falta de información de algunos genes de la cepa AG1. Es necesario encontrar genes en transcriptomas existentes que sean similares a los de la bacteria AG1, para así relacionar la información existente con los genes de esta bacteria que carecen de ella.
Con el algoritmo de Shingling, se busca encontrar similitud entre genes para agruparlos y permitir un análisis, más detallado de esos grupos, con algoritmos especializados existentes más precisos, como lo es el alineamiento global Needleman-Wunsch (NW).
Seguidamente se presenta información de antecedentes, el algoritmo de shingling mejorado para la búsqueda, comparación y asociación de genes junto con un experimento preliminar.

II. Antecedentes

El procesamiento de datos masivos requiere de algoritmos que optimicen tareas de búsqueda y comparación para obtener resultados en un tiempo computacional determinístico. Seguidamente se hace referencia al algoritmo de Shingling, aplicado para detectar documentos similares que afectan el proceso de indexación de estos en la Web. Además, se presenta el algoritmo de alineamiento global Needleman-Wunsch, usado en Bioinformática para el alineamiento y comparación de secuencias de ADN.

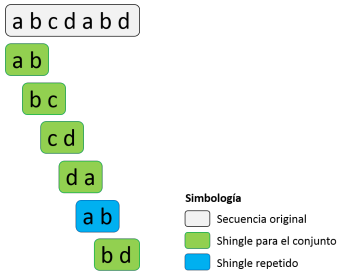

Figura 1. 2-Shingling de la secuencia *"abcdabd"*

II-A. Shingling

El autor del algoritmo Shingling es Andrei Z. Broder, quien da una solución para detectar documentos duplicados en la Web que afectan el proceso de indexación de los mismos. El algoritmo hace referencia al concepto de semejanza entre dos documentos y es una medida cuantificable entre los valores de 0 y 1. El valor 1 indica que dos documentos son exactamente el mismo. Al calcular la semejanza se puede mantener un *"sketch"*, que consiste de una colección de *"fingerprints"* (huellas) de shingles. El sketch puede ser calculado linealmente con respecto al tamaño de los documentos [2].

Cada documento se divide en una secuencia de tokens que representan letras, palabras o líneas.
Un shingle es una subsecuencia contigua de **w** tokens contenidos en un documento **D**.
El w-shingling de un documento **D** se define como el conjunto S_D de todos los singles de tamaño **w** contenidos en **D**.

Por ejemplo, el 2-shingling del documento **D** que contiene la secuencia *"abcdabd"* es el conjunto {ab, bc, cd, da, bd} [3]. La forma de obtener los 2-shinglig de *"abcdabd"* se muestra en la Figura 1.
Tomando en cuenta los shingles repetidos, la cantidad de shingles de una secuencia se obtiene de la siguiente forma:

$$(tamaño\ de\ la\ secuencia) - w + 1 \quad (1)$$

En el caso de la secuencia *"abcdabd"*, la cantidad de shingles se obtiene de la siguiente forma: **7 - 2 + 1 = 6**

La similitud r(A, B) de dos documentos A y B, que corresponde con el Coeficiente de Jaccard J(S_A, S_B) [4] se define como [2]:

$$r(A,B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \quad (2)$$

En [4] se hace un análisis del algoritmo de shingling y en [3] se hace una propuesta para generar una representación de cada documento, la cual es una huella llamada *"signature"*. Seguidamente se presentan los pasos del proceso propuesto en [3] para obtener los signatures que representan los documentos.

1) Hashing: La aplicación de una función de hash a los shingles permite mapear strings de tamaño **k** a algunos *"buckets"*, un bucket tiene asociado un valor del codominio de la funcón de hash, para el caso de la función h_1 = x + 1 mod 5, se tienen 5 buckets a donde se mapearn los shingles. El conjunto que representa el documento son los buckets de uno o más k-shingles que aparecen en él [3].

2) Signatures: La cantidad de shingles asociados a un documento es bastante grande, un inconveniente es el espacio necesario para almacenar la información. El signature es la representación para grandes conjuntos. Una propiedad es la posibilidad de comparar dos signatures y estimar la similitud de Jaccard [3].

3) Matriz característica: Es posible construir signatures de tamaño más pequeño cuando se tienen muchos conjuntos por medio de una matriz característica, la cual es binaria. Las columnas representan los conjuntos y las filas representan los elementos del conjunto universal contenidos en los conjuntos. Hay un 1 en la fila **r** y la columna **c** si el elemento para la fila **r** es miembro del conjunto para la columna **c**. Para otro caso el valor en la posición (r, c) será 0 [3].
En la Tabla I se muestra un ejemplo de una matriz característica que representa los conjuntos S_1 = {a, d}, S_2 = {c}, S_3 = {b, d, e}, S_4 = {a, c, d}, tomados del conjunto universal {a, b, c, d, e} [3].

Tabla I. MATRIZ CARACTERÍSTICA CON 4 CONJUNTOS

Elemento	S_1	S_2	S_3	S_4
a	1	0	0	1
b	0	0	1	0
c	0	1	0	1
d	1	0	1	1
e	0	0	1	0

4) Minhashing: Los signatures se componen de muchos resultados y cálculos, cada uno es un minhash en la matriz caracterástica. Un minhash es representado por una columna de la matriz característica y es producto de la permutación de las filas. El minhash de una columna es el número de la primera fila, en el orden permutado, cuyo valor es 1 [3].

5) Minhashing y Similitud de Jaccard: Existe una relación entre minhashing y la similitud de Jaccard de los conjuntos a los cuales se les ha obtenido el minhash [3]:

La probabilidad que la función de minhash para una permutación aleatoria de filas produzca el mismo valor para dos conjuntos, es igual a la similitud de Jaccard de esos conjuntos.

Si se tienen las columnas en la matriz característica de los conjuntos S_1 y S_2, las filas pueden ser de tres tipos:

a) X: las filas tienen un 1 en ambas columnas
b) Y: una fila tiene un 1 en una columna y un 0 en la otra
c) Z: las filas tienen un 0 en ambas columnas

Para determinar la similitud se consideran las filas de tipo X y Y, así la similitud se puede calcular de la siguiente forma:

$$SIM(S_1, S_2) = \frac{x}{(x+y)} \quad (3)$$

Donde S_1 y S_2 son conjuntos, x es la cantidad de filas de tipo X e y es la cantidad de filas de tipo Y. En el cálculo, x corresponde con $|S_1 \cap S_2|$ y (x + y) corresponde con $|S_1 \cup S_2|$ [3].

6) Cálculo de Minhash Signatures: La permutación aleatoria se puede obtener con una función de hash que mapea números de fila a tantos buckets como filas existan. Una función que mapea enteros de 0...k-1 a buckets numerados de 0...k-1, es posible que se mapeen 2 enteros en un mismo bucket y que otros buckets queden vacíos.
Para el cálculo se escogen n funciones hash h_1, h_2,..., h_n en las filas. Se construye la matriz signature considerando el orden de cada fila. SIG (i, c) es el elemento de la matriz de signature para la i-sima función de hash y columna c. Inicialmente se asigna ∞ a SIG(i, c) para todo i y c, como se muestra en la Tabla II.

Tabla II. MATRIZ DE SIGNATURES

	S_1	S_2	S_3	S_4
h_1	∞	∞	∞	∞
h_2	∞	∞	∞	∞

Para cada fila se hace lo siguiente:

a) Calcular h_1(r), h_2(r),..., h_n(r)
b) Para cada columna se hace lo siguiente:
   - Si c tiene 0 en la fila r, no hacer nada.
   - Si c tiene 1 en la fila r, entonces para cada i = 1, 2,..., n, SIG (i,c) será el valor mínimo entre SIG(i,c) y h_i(r).

Tabla III. CÁLCULO DE SIGNATURES - FUNCIONES HASH

Fila	S_1	S_2	S_3	S_4	h_1 = x + 1 mod 5	h_2 = 3x + 1 mod 5
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

El cálculo de los signatures busca el menor valor obtenido con cada función de hash[1], considerando la matriz característica mostrada en la Tabla III, donde las columnas representan los conjuntos y las filas representan los elementos del conjunto universal contenidos en los conjuntos, solo se toman en cuenta las filas que tienen unos para cada conjunto. Así para S_1, considerando solamente las filas con unos (filas 0 y 3), se obtiene la matriz mostrada en la Tabla IV, donde en la columna correspondiente a h_1, el menor valor es 1 y en la columna correspondiente a h_2, el menor valor es 0, las celdas se marcan con color verde. Estos valores son los que se consideran en el signature y se incluyen en la matriz de signatures, como se muestra en la Tabla V. El signature que representa el conjunto S_1 es 10.

[1]El procedimiento completo para obtener los signatures se puede consultar en [3], páginas 84 - 86.

Tabla IV. ANÁLISIS DEL CONJUNTO S_1

Fila	S_1	h_1 = x + 1 mod 5	h_2 = 3x + 1 mod 5
0	1	1	1
3	1	4	0

Tabla V. MATRIZ DE SIGNATURES

	S_1	S_2	S_3	S_4
h_1	1	3	0	1
h_2	0	2	0	0

II-B. Needleman-Wunsch (NW)

Saul Needleman y Christian Wunsch proponen un algoritmo de programación dinámica para el alineamiento óptimo de 2 secuencias v y w de longitudes m y n. La programación dinámica encuentra la solución a un problema inicial, valiéndose de soluciones a subproblemas más simples [5].
El algoritmo NW usa una tabla A de (n+1) por (m+1) celdas, cada celda A[i, j] almacenará el scoring de alineamiento óptimo de los prefijos v[0...i] y w[0...j] tomados respectivamente de v y w.
Eventualmente se deben agregar gaps o *"huecos"* en las secuencias para igualar la longitud de los prefijos procesados. NW razona que sólo hay 3 posibles formas de alinear los prefijos tomados de v y w:

a) Alinear óptimamente v[0...i-1] con w[0...j-1] y luego alinear el símbolo v[i] con el símbolo w[j] (ya sea con match o missmatch), o
b) Alinear óptimamente v[0...i-1] con w[0...j] y luego alinear el símbolo v[i] con un gap insertado en w después de w[0...j], o
c) Alinear óptimamente v[0...i] con w[0...j-1] y luego alinear el símbolo w[j] con un gap insertado en v después de v[0...i].

Se debe seleccionar el máximo scoring entre los tres, lo cual se puede expresar en la siguiente ecuación:

$$A[i][j] = max \begin{cases} \text{A}[i-1][j-1] + F(v[i], w[j]) \\ \text{A}[i][j-1] + G \\ \text{A}[i-1][j] + G \end{cases} \quad (4)$$

Donde F es una función que retorna M si ambos símbolos son iguales (match) o M' si son diferentes (missmatch). El valor de A[i][j] se calcula usando una de las 3 casillas previas, NW une la casilla actual con la casilla usada por medio de una línea. El alineamiento final se lee desde A[n][m] hasta A[0][0], siguiendo las líneas que unen las celdas, pueden haber varias soluciones o alineamientos y todas son óptimas [5].
Es necesario asignarle valores a G, M y M', una posible asignación es: G = 2, M = 1 y M' = -1.
Este algoritmo requiere espacio y tiempo cuadrático para su ejecución.

III. SHINGLING MEJORADO PARA BÚSQUEDA, COMPARACIÓN Y ASOCIACIÓN DE GENES

Se han incorporado mejoras en el algoritmo propuesto por [3], este rediseño considera las características existentes en la biología molecular, donde se tienen bases nitrogenadas que componen el ADN y el ARN, representadas por un conjunto particular de caracteres, a saber, A (adenina), T (timina), G (guanina), C (citosina) y U (uracilo). Además, se incluye el caracter comodín N (base nitrogenada desconocida).
Seguidamente se describen las mejoras incorporadas:

a) Se traduce cada posible shingle en un valor numérico para representarlo de forma única. Todos los posibles shingles se asocian con un número natural para asociarlo con el número de fila que ocupa en la matriz característica (sección II-A3). La traducción a valor numérico consiste en hacer un cambio de base, para este caso se tiene que el alfabeto es el conjunto {A, T, G, C, U, N}, lo que hace conveniente trabajar con valores en base 6. El valor final es el resultado de representar el shingle, que está en una base 6, en base 10. En la Tabla VI se muestran los valores numéricos asignados a cada símbolo del alfabeto.

Tabla VI. VALORES NUMÉRICOS ASIGNADOS A CADA SÍMBOLO (BASES NITROGENADAS)

Símbolo	A	T	G	C	U	N
Valor Numérico	0	1	2	3	4	5

b) La traducción de los shingles a valores numéricos permite prescindir de la matriz característica, ya que el valor asociado en base 10 corresponde al número de fila en la que existe y debería marcarse con un *"1"* dentro de dicha matriz. El beneficio es que solamente se realizan cálculos con los shingles que pertenecen a un gen específico.
c) Es posible aplicar las funciones de hash definidas (sección II-A6), a cada valor que representa un shingle sin necesidad de almacenar información, se almacena solo el menor valor obtenido al aplicar la función de hash y se actualiza la huella o signature, esto es si el valor obtenido es menor al actual.

También es necesario controlar las posibles colisiones que puedan ocurrir al usar las funciones de hash.

III-A. Selección de valores para agrupar signatures

Es importante considerar que el porcentaje de colisiones que se genera con las funciones de hash, indirectamente permitirá la agrupación y almacenamiento de signatures iguales en un lo que se llamaremos meta-bucket, al que se le asignará un identificador idéntico al de los signatures que contiene. Con esto se garantiza que los genes asociados a esos signatures poseen un cierto grado de similitud.
Para alcanzar la meta, se revisa en cuáles meta-buckets se ubican los signatures de los genes que carecen de información de la Pseudomona AG1 y se analizan de forma más precisa con el algoritmo Needleman-Wunsch (sección II-B). Así es posible inferir o asociar información a genes que carecen de ella. La Figura 2 muestra una representación de un meta-bucket.

La tarea más importante es seleccionar los valores que generen un porcentaje determinado de colisiones al aplicar las funciones de hash, es necesario parametrizar los siguientes valores para controlar el porcentaje de colisiones:

a) w o tamaño de los shingles
b) Cantidad de buckets de salida para las funciones de hash

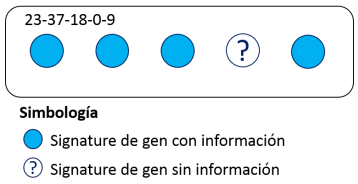

Figura 2. Meta-bucket

c) Tamaño del signature, o lo que es igual, la cantidad de funciones de hash a usar.

Lo que se busca es parametrizar valores que generen un porcentaje de colisiones determinado, que podría ser 25 % de colisiones o menos. Esta parte está aún en una etapa de investigación.

IV. Experimento Preliminar y Resultados

Se realizó un experimento preliminar de tipo factorial con 2 factores: el tamaño de los shingles o W (con valores 10, 20 y 30) y la cantidad de buckets (con valores 10000, 20000 y 30000), en total se tienen 9 iteraciones. Además, el tamaño de cada signature es 5, lo que implica el uso de 5 funciones de hash.
Las características de los datos usados en el experimento son las siguientes: 5163887 genes, extraídos de 963 cepas, cada cepa con alrededor de 5 KGenes. La informacin de cada gen est contenida en un archivo en formato fasta, el cual consta de un encabezado con datos del gen y luego la secuencia de bases nitrogenadas.

IV-A. Experimentación

El ambiente de experimentación consiste de un servidor de 32 cores y 64 GB de RAM. Se crea una aplicación que ejecuta el algoritmo de Shingling, cuya entrada consta de un archivo en formato fasta con informcación de un gen, el tamaño de W y la cantidad de buckets. Todo el proceso se realiza usando paralelismo con un script batch, se procesa cada archivo sin excluir ningún dato. El signature obtenido se asocia al archivo y se almacena en el correspondiente Meta-bucket.
Para verificar la similitud en los genes (archivos) asociados a cada Meta-bucket, se seleccionan 100 Meta-buckets de forma aleatoria y se les aplica el algoritmo de alineamiento global Needleman-Wunsch internamente (todos contra todos).
Se tomaron 1000 pares de muestras aleatorias, cada par se compararó usando el algoritmo Needleman-Wunsch. El promedio de similitud fue de 56 %.

IV-B. Resultados

Al aplicar el algoritmo de NW para verificar la similitud de forma puntual en los Meta-buckets se hace un análisis de varianza (ANOVA), los datos obtenidos se muestran en las tablas VII, VIII, IX y X.

V. Conclusión

Con el shinglig mejorado se busca inferir información para asociarla a genes que carecen de ella, particularmente a los genes de la Pseudomona AG1 con esta condición. Este algoritmo permite hacer un agrupamiento de genes en meta-buckets, se identifican los que contienen genes sin información y se analizan con el algoritmo Needleman-Wunsch. El agrupamiento propuesto evita la necesidad de comparar los signatures y determinar el grado de similitud de los genes, aunque se puede recurrir a la similitud de Jaccard. El proceso con shingling mejorado se realiza en tiempo de ejecución lineal y es posible el procesamiento con paralelismo.

Tabla VII. ANOVA: W = 10

	B=10000	B=20000	B=30000	Total
Cuenta	20097	20097	20097	60291
Suma	1328.18	13764.48	13808.41	40860.07
Promedio	0.66	0.68	0.69	0.68
Varianza	0.05	0.06	0.06	0.06

Tabla VIII. ANOVA: W = 20

	B=10000	B=20000	B=30000	Total
Cuenta	20097	20097	20097	60291
Suma	12393.08	13060.54	12890.68	38344.31
Promedio	0.62	0.65	0.64	0.64
Varianza	0.05	0.06	0.05	0.05

Tabla IX. ANOVA: W = 30

	B=10000	B=20000	B=30000	Total
Cuenta	20097	20097	20097	60291
Suma	14792.11	16445.71	16943.00	48180.82
Promedio	0.74	0.82	0.84	0.80
Varianza	0.05	0.05	0.05	0.05

Tabla X. ANOVA

Origen Variaciones	Suma Cuadrados	Grados Libertad	Promedio Cuadrados	F	Probabilidad	Valor Crítico F
Muestra	866.24	2	433.12	8081.20	0.00	3.00
Columnas	99.60	2	49.80	929.22	0.00	3.00
Interacción	46.87	4	11.72	218.65	0.00	2.37
Dentro Grupo	9693.58	180864	0.05			
Total	10706.30	180872				

Agradecimiento

Los autores desean agradecer a Jesús Ulate Cárdenas por su valiosa y desinteresada colaboración.

Referencias

[1] NCBI, *National Center for Biotechnology Information)*. NCBI. http://www.ncbi.nlm.nih.gov

[2] A. Broder, *Identifying and Filtering Near-Duplicate Documents*. Berlin Heidelberg: Springer-Verlag, 2000.

[3] J. Leskovec, J Rajaraman and J.D. Ullman, *Mining of Massive Datasets*. California, USA: Cambridge University Press, 2014.

[4] C. Manning, R. Prabhakar, H. Schutze, *An Introduction to Information Retrieval*. England: Cambridge University Press Cambridge, 2009.

[5] F. J. Torres-Rojas, *Algoritmos Clásicos de Alineamiento de Secuencias*. Costa Rica: Tiempo Compartido 7(1), 13-18, 2007.