Universidad de Concepción
Facultad de Ingeniería
Departamento de Ingeniería Informática y Ciencias de la Computación

ESTIMACIÓN DE DISTANCIA JENSEN-SHANNON USANDO SKETCHES APLICADA EN ANÁLISIS GENÓMICO

Informe de Memoria de Título para optar al título de Ingeniero Civil Informático

Autor: Juan Guillermo Albornoz Araya

Profesor patrocinante: Cecilia Hernández Rivas
Miembros de la comisión:
Lilian Salinas A.
Guillermo Cabrera V.

Septiembre de 2022

Sumario

El análisis y la comparación de secuencias genómicas es un área de investigación que va de la mano con la implementación de nuevos métodos capaces de lograr la computación de conjuntos de datos cada vez más masivos. Por esta razón, existe le necesidad de implementar y evaluar algoritmos y estructuras de datos que soporten la comparación de la información genómica de una manera más eficiente. En este contexto, se evalúa la implementación del uso de sketches, que son estructuras de datos probabilísticas con la ventaja de ser más eficientes que las estructuras convencionales. Los sketches aceptan consultas rápidas con baja complejidad lineal y espacio sublineal, no obstante, están sujetos a un error de estimación.

El trabajo [9], en el cual está basada esta memoria de título propone un método de estimación de la Entropía empírica de Shannon, sobre tráfico de flujo datos en redes. El método utiliza algoritmos basados en sketches, para optimizar el uso de los recursos, y el cual se implementa sobre una arquitectura de hardware especialmente diseñada para el método de estimación de la Entropía de Shannon.

Esta memoria de título presenta una propuesta de estimación de la métrica Distancia Jensen-Shannon, basada en la estimación de la Entropía de Shannon usando sketches propuesta en [9]. La métrica de distancia es evaluada en el contexto de la comparación de similitud entre distintos conjuntos de secuencias genómicas, para lo cual se realiza una búsqueda de los parámetros óptimos mediante la experimentación y el análisis de los resultados de estimación de entropía. Los resultados obtenidos son positivos, por un lado, en relación con la precisión de estimación de la métrica, y también en cuanto al uso de recursos de memoria utilizados.

Índice general

Índice de tablas

Índice de figuras

Índice de algoritmos

Capítulo 1

Introducción

La genómica computacional, ligada al área de la bioinformática, plantea el uso de distintas técnicas algorítmicas, de análisis estadístico y Machine Learning, para resolver los problemas en el área. Desde comienzos de la década de 1980, los procesos de secuenciación masiva de genomas han reducido sus costos, lo que ha producido un crecimiento continuo en los volúmenes de datos. La gran cantidad de datos disponibles ha motivado el desarrollo de nuevos algoritmos de comparación de secuencias que buscan abordar la solución de problemas en esta área. Estos algoritmos han permitido, por ejemplo, la identificación de regiones de similitud entre distintos genomas de ADN, dando paso al análisis y comparación de sus rasgos estructurales, funcionales, o evolutivos. Los genomas de ADN se definen en base a la combinación de las bases de nucleótidos A, C, T y G, las cuales forman una secuenciación que caracteriza de manera única algún organismo en particular.

Existen métodos tradicionales y pioneros que abordan la búsqueda de similitud entre la secuenciación de distintos genomas de ADN, los cuales pueden ser del tipo dependientes de alineación, o libres de alineación. La alineación [1] consiste básicamente en la organización de determinadas secuencias de distintos genomas, de manera tal de posibilitar, por ejemplo, la comparación e identificación de regiones de similitud entre estos genomas. Los métodos dependientes de alineación entregan buenos resultados, pero están limitados por su complejidad computacional y tiempo de ejecución al ser utilizados en la comparación de genomas completos. El algoritmo $BLAST$ [2], por ejemplo, el cual es dependiente de alineación, no fue diseñado para la comparación de genomas completos. $BLAST$ intenta buscar secuencias coincidentes de manera exacta entre los genomas, y debido a que la comparación par a par es computacionalmente costosa, este resultaría en tiempos de ejecución en el orden cuadrático [3].

"La genómica a gran escala exige métodos computacionales que escalen con el crecimiento de los datos" [4], debido a esto, y como alternativa a los métodos dependientes de alineación, se han propuesto diversos métodos libres de alineación [4]. Estos métodos se subdividen en distintos grupos, y uno de los más populares se define en base al conteo de las secuencias genómicas de un determinado largo k, denominados k-mer, el cual se utiliza en esta memoria, y es descrito en una sección posterior. Los métodos libres de alineación son menos costosos computacionalmente en comparación a métodos dependientes de alineación, pero pueden llegar a requerir un uso elevado de memoria cuando se trabaja sobre grandes volúmenes de datos genómicos.

Se han propuesto, además, diversas métricas de Teoría de la Información, por ejemplo, la Entropía de Shannon y la Divergencia Jensen-Shannon [5], que buscan entregar información respecto a la diversidad o aleatoriedad de un conjunto de datos, esto en base a distribuciones probabilísticas. Estas métricas, las cuales se describirán más en profundidad en la sección siguiente son la base del desarrollo de esta memoria, y, además, su uso en el

contexto de análisis genómico no es nuevo. Existen métodos de comparación de secuencias genómicas, del tipo libres de alineación, los cuales se basan en la Entropía de Shannon, y han sido adaptados para la construcción de árboles filogenéticos capaces de inferir relaciones entre organismos [6]. También, la Divergencia Jensen-Shannon, la cual es calculada en términos de la Entropía de Shannon, se ha utilizado, por ejemplo, en la elaboración de métodos heurísticos capaces de detectar interacciones asociadas a múltiples enfermedades [7], o en la medición de distancia entre diferentes secuencias genómicas mediante el uso de la técnica de Perfil de Frecuencia de Características o FFP (Feature Frequency Profile) [8], la cual cabe dentro del grupo de los métodos libres de alineación nombrado anteriormente.

Un trabajo implementa un método de estimación de la Entropía empírica de Shannon, para el análisis de tráfico en redes [9], mediante algoritmos que utilizan estructuras de datos probabilísticas denominadas sketches [10]. Un sketch es una estructura de datos reducida en espacio, usada para mantener datos que se utilizan para evaluar y/o calcular una estimación de una función de interés. Los sketches permiten generar un conjunto de datos más compacto que el original, aceptan consultas con baja complejidad lineal y con espacio sub-lineal, lo que permite optimizar el uso de los recursos, pero están sujetas a un error de estimación. Son utilizados para diversas aplicaciones donde se dispone de pocos recursos de memoria o se desea implementar la resolución de consultas en línea, y son aplicados también, por ejemplo, en el contexto de procesamiento de señales, en problemas de reducción de dimensionalidad, entre otros. El uso de sketches en el contexto del análisis genómico tampoco es nuevo, sin embargo, su aplicación junto al método de estimación de la Entropía empírica de Shannon propuesto en [9] si lo es.

En esta memoria se propone adaptar el método de estimación de la Entropía empírica de Shannon, usando algoritmos basados en sketches, y evaluar su factibilidad en la estimación de la Distancia Jensen-Shannon sobre un conjunto de genomas disponibles en la base de datos RefSeq de NCBi (National Center for Biotechnology Information) [11], y de esta manera tener una métrica de similitud entre los genomas de ADN.

1.1 Objetivo general

Proponer e implementar un método para el cálculo de la métrica Distancia Jensen-Shannon, usando estimación de entropía basada en sketches, en el contexto de la comparación de conjuntos de secuencias genómicas (Se recomienda dirigirse al glosario para aclarar el concepto de conjunto de secuencias genómicas).

1.2 Objetivos específicos

a) Adaptar los algoritmos y estructuras de datos utilizados en el método propuesto en "A High-Throughput Hardware Accelerator for Network Entropy Estimation Using Sketches" [9], para su utilización en la estimación de la entropía empírica de Shannon sobre conjuntos de secuencias genómicas.

b) Implementar el método propuesto de estimación de entropía empírica de Shannon, para ser utilizado sobre conjuntos de secuencias genómicas.

c) Definir e implementar un método de estimación de la métrica Distancia Jensen-Shannon, mediante el uso de la estimación de la entropía empírica de Shannon, con el propósito de comparar la similitud entre conjuntos de secuencias genómicas.

d) Evaluar el desempeño del método de estimación de la Distancia Jensen-Shannon, mediante la comparación con los resultados exactos de esta métrica.

Capítulo 2

Marco teórico

En esta sección se describe el método propuesto en [9] de estimación de la Entropía empírica de Shannon, en el contexto del análisis de tráfico de paquetes de redes, por lo cual se describen los algoritmos utilizados en el proceso, y las estructuras de datos involucradas. Esto es relevante, debido a que la estimación de la métrica Distancia Jensen-Shannon es realizada en base a la Entropía empírica de Shannon, y es esencial explicar el funcionamiento del método planteado y sus algoritmos. De aquí en adelante se utiliza simplemente el término entropía para la estimación de la Entropía empírica de Shannon, y también, se utiliza el término elemento para hacer referencia al dato de red que se utiliza, pudiendo ser, por ejemplo, IP origen, IP destino, puerto, etc.

La entropía en Teoría de la Información es la medida de incertidumbre de una variable aleatoria, por lo tanto, mientras más diversos o aleatorios sean los valores de alguna distribución (Ver glosario) mayor es la entropía. Realizar el cálculo exacto de la entropía es desafiante, ya que requiere mantener una gran cantidad de contadores, esto es, uno para cada elemento distinto, además de requerir consultar esos contadores a velocidades que dependen del flujo de elementos (Ver glosario). El trabajo base referenciado [9] propone un método para estimar la entropía, esto en el contexto de tráfico de alta velocidad de dispositivos de redes, esto con memoria limitada y bajos recursos de procesamiento. Para esto, los algoritmos basados en sketches cumplen un rol fundamental, ya que permiten generar un conjunto de datos más compacto, aceptan consultas con baja complejidad lineal y espacio sub-lineal. Los algoritmos basados en sketches utilizados son HyperLogLog sketch para la estimación de cardinalidad (Ver glosario), tres distintos algoritmos para la estimación del conteo de frecuencias de los elementos, y una cola de prioridades aproximada que almacena los elementos más frecuentes. El método de estimación de entropía, los algoritmos y sus estructuras involucradas se describen en detalle a lo largo de esta sección.

2.1. Método propuesto de estimación de Entropía empírica de Shannon

El método de estimación de entropía propuesto e implementado en [9], se evalúa utilizando algoritmos basados en sketches, sobre una arquitectura de hardware especialmente diseñada para lograr un alto rendimiento en el procesamiento del flujo de los elementos. En el trabajo se transforma el cálculo exacto de la entropía H (2.1) y la entropía normalizada H_n (2.2) en fórmulas que estiman la entropía H. La Entropía H (2.1) depende del conteo de la frecuencia de aparición del elemento denotada por m_i ,$i \in \{1, \dots, N\}$, también de la cardinalidad (Ver glosario) del flujo de elementos denotada por N, y del contador de los elementos totales denotado por M.

$$H = -\left[\sum_{i=1}^{N} \frac{m_i}{M} * log_2\left(\frac{m_i}{M}\right)\right] \tag{2.1}$$

$$H_n = \frac{H}{log_2(N)} \tag{2.2}$$

El enfoque establecido en el trabajo [9] es que, al calcular la entropía de un conjunto de datos masivo, los elementos con mayor frecuencia aportan una mayor contribución a la entropía que los elementos con menor frecuencia. En base a esto, se transforma el cálculo exacto de la entropía H a un método de estimación en el cual se separa la contribución, por un lado, de los elementos más frecuentes, y por otro lado la contribución del resto de los elementos. Es decir, la fórmula (2.1) se separa en las expresiones $left$ (2.3) y right (2.4). Por un lado, $left$ considera la estimación de entropía de la cantidad K de elementos más frecuentes, denominados top-K (Ver glosario), y $right$ considera la estimación de entropía de una distribución uniforme para los elementos restantes. El término $right$ depende de la cardinalidad N del flujo de elementos, además de la suma acumulada de los K elementos más frecuentes, la cual se denota por L, y la cantidad K de elementos más frecuentes considerados en la estimación. Nótese la diferencia con el término k minúscula, que es utilizada en la denotación de los términos k-mer y top-K, los cuales se utilizan en la sección posterior de desarrollo de este informe.

$$left = \sum_{i=1}^{K} \frac{m_i}{M} * log_2\left(\frac{m_i}{M}\right) \tag{2.3}$$

$$right = \left[\frac{M-L}{M} * log_2\left(\frac{M-L}{M*(N-K)}\right)\right] \tag{2.4}$$

Es decir, la estimación de la entropía H (2.1) se realiza en base a la expresión (2.5) que equivale a la suma de los términos anteriores (2.3) y (2.4), y es denotada por $\hat{H}$:

$$\hat{H} = -\left[\sum_{i=1}^{K}\left(\frac{m_i}{M} * log_2\frac{m_i}{M}\right) + \frac{M-L}{M} * log_2\left(\frac{M-L}{M*(N-K)}\right)\right] \tag{2.5}$$

A continuación, se simplifican cada uno de los sumandos, y de esta manera se evita la división por el contador M en cada una de las iteraciones del operando sumatoria del término $left$. El término derecho también se simplifica y es factorizado por el término $-\frac{1}{M}$. La expresión (2.6) resultante es la siguiente:

$$\hat{H} = -\frac{1}{M}\left[\left(\sum_{i=1}^{K} m_i * log\,(m_i) - L * log\,(M)\right)\right] -$$

$$\frac{1}{M}\left[(M-L) * \left(\log(M-L) - log\big(M * (N-K)\big)\right)\right] \quad (2.6)$$

2.2. Algoritmos basados en sketches

Los algoritmos utilizados son, HyperLogLog sketch [12] para la estimación de la cardinalidad N, tres algoritmos de estimación de frecuencia; estos son, CountMin [13], CountMin Conservative Update sketch [13] y Countsketch [14]. Además, se implementa una cola de prioridades aproximada que almacena los elementos más frecuentes, la cual será descrita en una sección posterior.

2.2.1. HyperLogLog sketch

El algoritmo HyperLogLog sketch, abreviado como HLL, permite estimar la cantidad de elementos distintos de un flujo de datos o cardinalidad. HLL permite una actualización rápida y de gran precisión, comparado, por ejemplo, a métodos basados en MinHash [15].

El sketch se inicializa en base a un vector denotado por A, con un tamaño equivalente a $|A| = 2^{hll_p}$ (Algoritmo 2.1), en el que los buckets (Ver glosario) son inicializados en valor cero. El término denotado por hll_p es el parámetro de precisión del sketch, y está relacionado al tamaño del vector A. También, se define un factor de corrección denotado por α_A que depende del tamaño del vector A, y ayuda a corregir un sesgo multiplicativo en el cálculo posterior de la cardinalidad N. Esto es, para $|A| \geq 128$, α_A es calculado mediante una expresión que depende del tamaño del vector A. Sin embargo, para valores de $|A| < 128$ se consideran valores predefinidos de α_A.

Algoritmo 2.1 Inicialización de HyperLogLog sketch

Entrada: Parámetro de precisión *hll_p*

1: $|A| = 2^{hll\_p}$
2: $\alpha_A \leftarrow 0.723/(1+1.079/|A|)$, si $|A| \geq 128$
3: $A[i] \leftarrow 0$, $i \in [0, |A|\text{-}1]$

El contenido del vector siempre se actualiza en el momento que cada elemento es procesado, lo cual se explica a continuación con más detalle. El método de actualización del sketch (Algoritmo 2.2) recibe el valor hash de 32 bits, denotado por h, de un elemento del flujo, y a partir de su representación de bits se definen los valores de v_1 y de v_2 (Ver anexo 1). Para el valor de v_1 se consideran los hll_p bits más a la izquierda, desde la posición h_{31} hasta h_{32-hll_p}, los que se utilizan como índice de posición en el vector A. Para el valor de v_2,

Algoritmo 2.2 Actualización de HyperLogLog sketch

Entrada: Valor hash h(e) de elemento e (32bits)

1: $v_1 \leftarrow < h(e)_{31}, \dots, h(e)_{32-hll\_p} > 2$

2: $v_2 \leftarrow < h(e)_{31-hll\_p}, \dots, h(e)_0 > 2$

3: $A[v_1] \leftarrow máx\{A[v_1], ldz(v_2) + 1\}$

en cambio, se consideran los bits desde la posición h_0 hasta h_{31-hll_p}, y luego se cuentan los ceros más a la izquierda (Ver glosario), sumándole 1 al conteo $(leadingzeros(v_2) + 1)$. El valor resultante de esta operación se reemplaza en el bucket del vector A en la posición $A[v_1]$, sólo si el valor a almacenar es mayor al que está almacenado actualmente. La obtención de los valores v_1 y v_2 se describe gráficamente en la Figura 2.1 y se detalla en el Algoritmo 2.2. Nótese que en la Figura 2.1 se utiliza p en vez de hll_p en la descripción de los valores de v_1 y v_2, esto por motivos de facilidad en la notación del esquema.

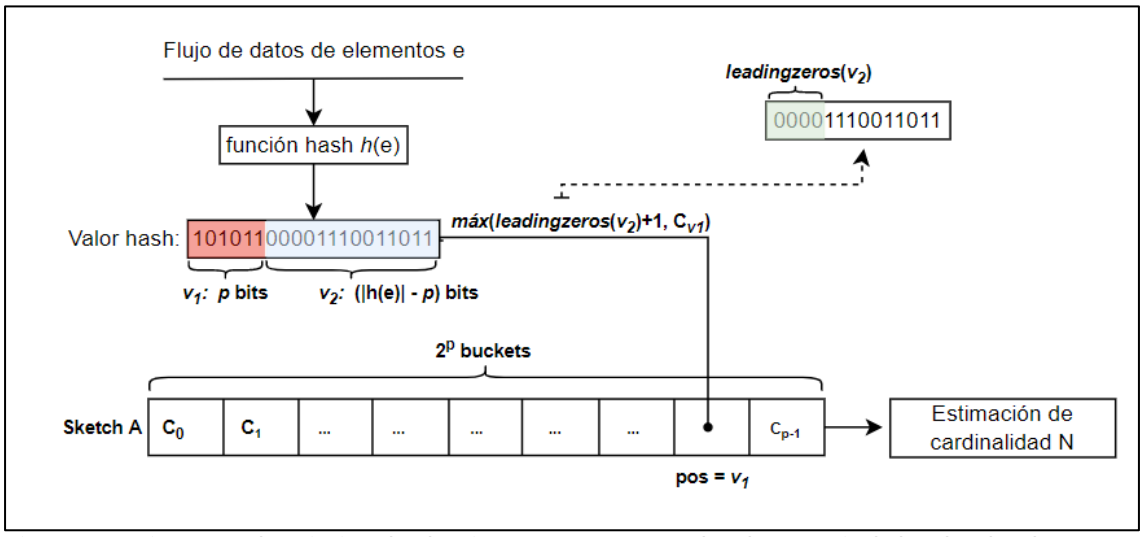


Figura 2.1: Diagrama descriptivo de algoritmo HyperLogLog Sketch. A partir del valor hash $h(e)$ se obtienen los valores v_1 y v_2. Luego, se calculan los leading zeros + 1 del valor v_2, y se reemplaza el resultado en el vector A en la posición $A[v_1]$ sólo si el valor a reemplazar es mayor al actual.

La estimación de la cardinalidad N del flujo de elementos (Algoritmo 2.3), se realiza una vez todos los elementos son procesados. En primer lugar, se calcula la media armónica de todos los contadores almacenados en el vector A denotada por Z, y al mismo tiempo se hace el conteo de los buckets de A con valor almacenado cero, el cual se denota por n_z. Finalmente, se evalúa si es que el valor del N estimado que resulta de la expresión $(\alpha_A * |A|^2/Z)$ es menor o igual a una cota equivalente a $(2.5 * |A|)$ [9], lo que permite juzgar si es conveniente realizar la estimación por un cálculo alternativo de la cardinalidad N equivalente a $(|A| * \log(|A|/n_z))$, el cual resulta en un menor error de estimación. Esto se relaciona con la cantidad de buckets con valor cero del vector A, que resultan al realizar la estimación a conjuntos de cardinalidad baja comparada al tamaño del vector A. Además, el

algoritmo tiene una complejidad espacial sublineal equivalente a O($|A| * loglog(N)$) y un error estándar de $1.03/\sqrt{|A|}$ [9].

Algoritmo 2.3 Estimación de cardinalidad N por HyperLogLog sketch

Salida: Cardinalidad N

1: $Z \leftarrow \sum_{i=0}^{|A|-1} 2^{-A[i]}$

2: $N_{HLL} \leftarrow \alpha_A * \frac{|A|^2}{Z}$

3: $\boldsymbol{si}\ N_{HLL} \leq 2.5 * |A|\ \boldsymbol{entonces}$

4: $\quad n_z \leftarrow ContCeros(A)$

5: $\quad N_{HLL} \leftarrow |A| * log_2(|A|/n_z)$

6: $\boldsymbol{fin\ si}$

7: $\boldsymbol{devolver}\ N_{HLL}$

2.2.2. CountMin, CountMin Conservative Update y Countsketch

Los algoritmos CountMin (CM), CountMin Conservative Update sketch (CM-CU), y Countsketch (CS) son algoritmos basados en sketches, los cuales permiten estimar la frecuencia de elementos de un flujo de datos. Son comúnmente utilizados, por ejemplo, en el problema de obtención de Heavy-Hitters (Ver glosario) o en el conteo de elementos de un flujo de datos en streaming.

Los sketches están conformados por una matriz denotada por C (Algoritmo 2.4), la cual se dimensiona mediante los parámetros de precisión denotados por $depth$, y $width$. El parámetro de precisión $depth$ determina la cantidad de filas, y $width$ corresponde a la cantidad de bits que dimensionan cada fila de buckets de la matriz C. Es decir, la cantidad de columnas (o buckets en cada fila de la matriz) equivale a 2^{width}, y cada una de las filas j hace referencia a una función hash d_j, la cual es utilizada para mapear el valor $h(e)$ del elemento e en un bucket de la fila j del sketch.

Algoritmo 2.4 Inicialización de CM-CU sketch, CM sketch y CS

Entrada: Parámetros width y depth. Funciones hash $d_j, j \in [0, \dots, depth-1]$

1: $C[j][i] \leftarrow 0, j \in [0, \dots, depth-1], i \in [0, \dots, 2^{width}-1]$

El método de inicialización de los sketches es igual para los tres algoritmos, sin embargo, presentan diferencias en el método de actualización y consulta. Se debe destacar que al igual que en el algoritmo HLL de estimación de cardinalidad, el contenido de la matriz siempre se actualiza en el momento que cada elemento es procesado, lo cual se explica a continuación con más detalle.

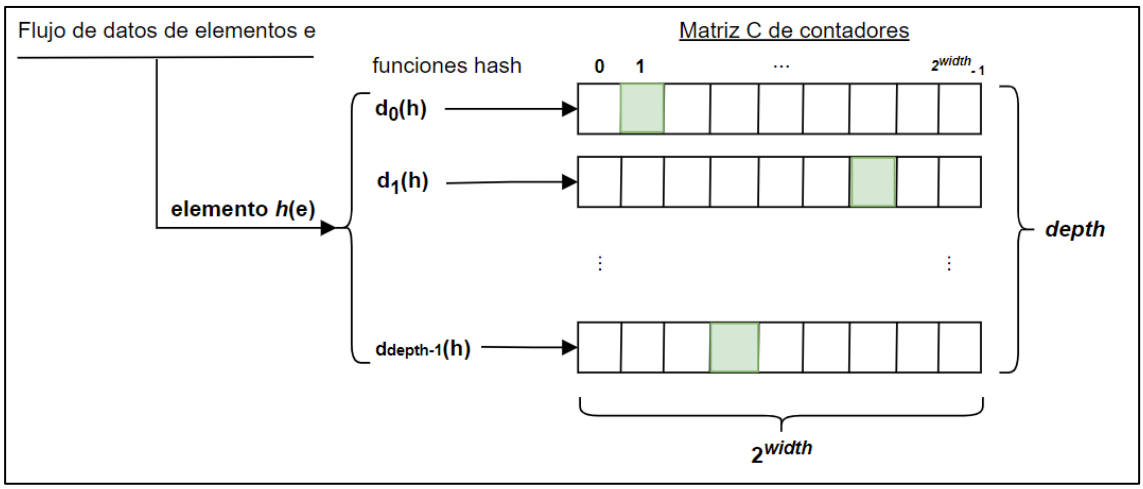


Figura 2.2: Diagrama descriptivo de la matriz de contadores de los algoritmos CountMin, CountMin-CU y Countsketch. El valor hash h(e) del elemento e es mapeado en la matriz C mediante $depth$ adicionales funciones hash independientes por pares.

De acuerdo con el método de actualización, se recibe el valor hash $h(e)$, y se mapea en cada fila de la matriz mediante las $depth$ funciones hash d_j, esto de la manera siguiente. En el caso del algoritmo CountMin (Algoritmo 2.5), se incrementan todos los contadores, esto es, todos los buckets en $C[j, d_j(h)], j \in [0, \dots, depth-1]$. Sin embargo, en el caso del algoritmo CountMin-CU se busca el mínimo de los contadores almacenados en los buckets de acuerdo con la fila j y la columna con el índice equivalente al valor hash $d_j(h)$, es decir, el $min\{C[j, d_j(h)]\}$, $j \in [0, \dots, depth-1]$, y se incrementan en 1 todos los contadores que son iguales al mínimo encontrado (Algoritmo 2.6). En el caso del algoritmo Countsketch, en cambio, se utilizan otras f_j funciones hash para decidir si incrementar o disminuir el contador (Algoritmo 2.7).

Algoritmo 2.5 Actualización de CountMin sketch

Entrada: Valor hash *h* de elemento e (32bits)

1: $\boldsymbol{para}\ j = 0\ to\ depth - 1\ \boldsymbol{hacer}$
4: $\quad C[j, d_j(h)] \leftarrow C[j, d_j(h)] + 1$
6: $\boldsymbol{fin\ para}$

Algoritmo 2.6 Actualización de CountMin Conservative update sketch

Entrada: Valor hash *h* de elemento e (32bits)

1: $\boldsymbol{para}\ j = 0\ to\ depth - 1\ \boldsymbol{hacer}$
2: $\quad min\_est \leftarrow min\{C[j, d_j(h)]\}, j \in [0, \dots, depth-1]$
3: $\quad \boldsymbol{si}\ C[j, d_j(h)] = \text{min\_est}\ \boldsymbol{entonces}$
4: $\quad\quad C[j, d_j(h)] \leftarrow C[j, d_j(h)] + 1$
5: $\quad \boldsymbol{fin\ si}$
6: $\boldsymbol{fin\ para}$

Algoritmo 2.7 Actualización de Countsketch

Entrada: Valor hash *h* de elemento e (32bits) y valor hash *f(h)* ϵ *{-1, 1}*

1: $\boldsymbol{para}\ j = 0\ to\ depth - 1\ \boldsymbol{hacer}$

4: $\quad C\left[j, d_j(h)\right] \leftarrow C\left[j, d_j(h)\right] + \{1 * f_j(h)\}$

6: $\boldsymbol{fin\ para}$

El método de estimación para un elemento *e* retorna el valor estimado de su frecuencia que equivale al mínimo de los contadores presentes en los buckets $C\left[j, d_j(h)\right]$, es decir, el $min\{C\left[j, d_j(h)\right]\}$, $j \in [0, \dots, depth - 1]$. Este método de estimación es el mismo en el algoritmo Countmin-CU y CountMin (Algoritmo 2.8), sin embargo, varía en el caso del algoritmo Countsketch (Algoritmo 2.9), ya que la obtención de la frecuencia estimada del elemento *e* se realiza mediante el cálculo de la mediana de los contadores $C\left[j, d_j(h)\right]$.

Algoritmo 2.8 Estimación de frecuencia de elemento e mediante CM-CU y CM sketch

Entrada: Valor hash *h* de elemento e (32bits)

Salida: Estimación de frecuencia *est* del elemento

1: $\boldsymbol{devolver}$ est $\leftarrow \mathbf{mínimo}\{C\left[j, d_j(h)\right]\}, j \in [0, \dots, depth - 1]$

Algoritmo 2.9 Estimación de frecuencia de elemento e mediante Countsketch

Entrada: Valor hash *h* de elemento e (32bits)

Salida: Estimación de frecuencia *est* del elemento

1: $\boldsymbol{devolver}$ est $\leftarrow \boldsymbol{mediana}\{C\left[j, d_j(h)\right]\}, j \in [0, \dots, depth - 1]$

2.2.3. Cola de prioridad de elementos más frecuentes

La estructura de la cola de prioridad aproximada es propuesta en el trabajo [9], y tiene el objetivo de obtener los elementos con frecuencias más altas. La implementación de la estructura presenta diferencias importantes si la comparamos con una cola de prioridad clásica, ya que se construye en base a múltiples colas más pequeña, y lo cual permite disminuir drásticamente el tiempo de ordenamiento de los elementos.

Está conformada por una matriz Q de vectores de pares (Figura 2.3), mediante los parámetros de precisión $height$ y $width$. El parámetro $height$ determina el número de filas, el cual equivale a 2^{height} (Algoritmo 2.10), y el parámetro $width$ determina el número de pares que puede almacenar como máximo cada fila, el cual equivale a 2^{width}. Esto quiere decir, que la cantidad de los elementos más frecuentes considerados depende del tamaño de la cola de prioridad. Cada par se conforma por un tag identificador, el cual corresponde al valor hash $h(e)$ de 32 bits del elemento, y por el valor estimado de la frecuencia del elemento a agregar a la cola. Se debe destacar que al igual que en el algoritmo HLL estimación de cardinalidad, y en el algoritmo CountMin de estimación de frecuencias, el

contenido de la matriz de la cola siempre se actualiza en el momento que cada elemento es procesado, lo cual se explica a continuación con más detalle.

Algoritmo 2.10 Inicialización de la de cola de prioridad

Entrada: Parámetros height y width.

1: $\boldsymbol{para}\ i = 0\ to\ 2^{height} - 1\ \boldsymbol{hacer}$
2: $\quad Q[i] \leftarrow vector < pair < int, int >>$
3: $\boldsymbol{fin\ para}$

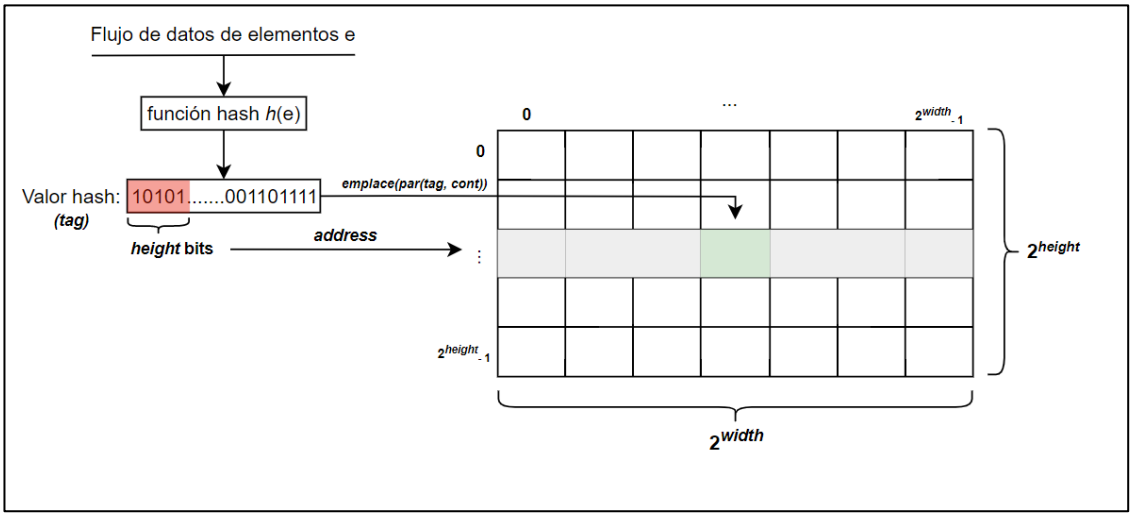


Figura 2.3: Diagrama descriptivo de la matriz de la cola de prioridad. La matriz C se dimensiona mediante los parámetros de precisión $height$ y $width$. Los pares se conforman por un tag equivalente al valor hash $h(e)$ de 32 bits del elemento, y el contador estimado de la $\mathrm{frecuencia}$ del elemento obtenido del algoritmo CountMin.

El índice $address$ se construye en base a los bits del valor h que van desde la posición del bit menos significativo hasta el bit en la posición $31 - height$. Este valor permite establecer la posición del vector en Q de la cola en la que se agregará el par formado por el identificador tag, y el contador estimado $cont$ de la frecuencia de dicho elemento, esto de la forma $pair < tag, cont >$.

Para añadir un elemento a la cola de prioridad se verifican 3 condicionales (Algoritmo 2.11). El primer caso consiste en que la cola del vector $Q[address]$ está vacía, y en tal caso se agrega el elemento directamente en la cola. El segundo caso hace referencia a que ya existen otros elementos agregados anteriormente en la cola, por lo que en tal caso se compara el tag del elemento candidato con el de los elementos ya presentes en la cola y se reemplaza el valor $cont$ en caso de encontrarse el mismo valor tag, si no, en cambio, solo se agrega un nuevo par. El tercer y último caso, consiste en que la cola del vector

$Q[address]$ se quedó sin espacio disponible, por lo que se debe realizar el mismo proceso anterior de comparación del tag, y reemplazar el valor $cont$ o agregar el par completo. Sin embargo, al sobrepasarse el límite de elementos permitidos en la cola del vector $Q[address]$, entonces es necesario realizar un ordenamiento, luego eliminar aquél par con el mínimo valor estimado de $cont$ almacenado, y de esta manera se conservan los elementos más frecuentes del flujo.

Algoritmo 2.11 Agregar elemento a la cola de prioridad

Entrada: Valor hash *h(e)* de elemento *e* (32bits) y contador *cont*

```
1:  address ← < h_{height−1}, … , h_0 >
2:  tag ← h
3:  si size(Q[address]) = 0 entonces
4:      Q[address] ← pair < tag, cont >
5:      devolver
6:  fin si
7:  si size(Q[address]) < 2^{width} entonces
8:      para j = 0 to size(Q[address]) hacer
9:          si Q[address][j].first = tag entonces
10:             Q[address][j].second ←cont
11:             devolver
12:         fin si
13:     fin para
14:     Q[address] ← pair < tag, cont >
15:     devolver
16: fin si
17: en otro caso entonces
18:     para j = 0 to size(Q[address]) hacer
19:         si Q[address][j].first = tag entonces
20:             Q[address][j].second ← cont
21:         fin si
22:     fin para
23:     Q[address] ← pair < tag, cont >
24:     sort(Q[address])
25:     pop_back(Q[address])
26: fin en otro caso
```

2.3. Estimación de entropía $\hat{H}$ de los elementos más frecuentes

Luego de realizarse la estimación de frecuencia de los distintos elementos mediante alguno de los algoritmos de estimación, sea este CountMin, CountMin-CU o Countsketch, y haberse almacenado las frecuencias de los elementos más frecuentes en la cola de prioridad, además de haberse calculado la cardinalidad N de un conjunto de elementos, es posible estimar la entropía H mediante $\hat{H}$.

Para obtener la entropía estimada $\hat{H}$, en primer lugar, se calcula la suma acumulada de los elementos más frecuentes almacenados en la cola, denotada por L, y la suma acumulada de la expresión factorizada equivalente a $\sum_{i=1}^{K} m_i * log(m_i)$ (Algoritmo 2.12), con m_i como la frecuencia estimada del elemento i, con $i \in \{1, \dots, K\}$, y K denota a la cantidad total de los elementos más frecuentes. Se debe hacer notar que el término $left$ utilizado en el algoritmo siguiente, es un término factorizado de la expresión (2.3), el cual se utiliza en la expresión final de entropía estimada $\hat{H}$ (2.6).

Algoritmo 2.12 Obtener término *left* de H y frecuencia acumulada L de elementos más frecuentes

Salida: Valores L, *left*

```
1:  L ← 0, left ← 0
2:  para address = 0 to 2^height − 1 hacer
3:      para j = 0 to size(Q[address]) hacer
4:          m_i ← Q[address][j].second
5:          L ← L + m_i
6:          left ← left + (m_i * log_2(m_i))
7:      fin para
8:  fin para
```

Finalmente, se reemplazan en la expresión (2.6) la suma acumulada L de los elementos más frecuentes, el contador de elementos totales M, la cardinalidad N, y la cantidad K de elementos más frecuentes.

$$\hat{H} = -\frac{1}{M}\left[\left(\sum_{i=1}^{K} m_i * log(m_i) - L * log(M)\right)\right] -$$

$$\frac{1}{M}\left[(M-L) * \left(\log(M-L) - log\big(M * (N-K)\big)\right)\right] \qquad (2.6)$$

2.4. Distancia Jensen-Shannon

La Divergencia de Jensen-Shannon es una versión simétrica de la Divergencia de Kullback-Leibler [16]. La Divergencia de Kullback-Leibler (2.7) es una medida no simétrica de la similitud entre dos funciones de distribuciones de probabilidad. La propiedad de simetría tiene relación a que la medida de la divergencia de una distribución A a una distribución B es equivalente a la medida de la divergencia de B a A. La divergencia se puede entender como una medida que cuantifica la diferencia entre distribuciones o poblaciones.

La Divergencia de Jensen-Shannon puede ser calculada en términos de Entropía H según la fórmula (2.7), y permite medir la desviación entre la Entropía H de la combinación de dos distribuciones A y B, denotada por $H(M_{AB})$, y el promedio de la Entropía H de las distribuciones A y B, definido por $-\frac{1}{2}\big(H(A)+H(B)\big)$.

La combinación de dos distribuciones, la cual se denota por M_{AB}, se define mediante la expresión (2.8), y la métrica comparativa, como tal, equivale a la raíz cuadrada de la Divergencia Jensen-Shannon, la cual se denomina Distancia Jensen-Shannon (2.9).

$$JSDiv = H(M_{AB}) - \frac{1}{2}\big(H(A)+H(B)\big) \tag{2.7}$$

$$M_{AB} = \frac{1}{2}(A+B) \tag{2.8}$$

$$JSDist = \sqrt{JSDiv} \tag{2.9}$$

Capítulo 3

Desarrollo

3.1. Estimación de la Distancia Jensen-Shannon

En este capítulo se detalla el método propuesto para la estimación de la Distancia Jensen-Shannon, según la fórmula (2.7), y su implementación para estimar la similitud entre conjuntos de secuencias genómicas, esto mediante el uso de la estimación de entropía $\widehat{H}$.

En el contexto del análisis genómico se había mencionado brevemente en la introducción un grupo de métodos libres de alineación basados en el conteo de las secuencias k-mer de genomas de ADN. Un método en específico denominado FFP (Feature frequency profile) [17], construye un vector conformado por el conteo de las frecuencias de las secuencias k-mer, el cual permite, por ejemplo, medir la Divergencia Jensen-Shannon entre distintos conjuntos de secuencias.

En la sección anterior se describió el método propuesto en [9] de estimación de la entropía $\widehat{H}$, el cual será utilizado de acuerdo con las fórmulas de cálculo de la métrica Distancia Jensen-Shannon. Sin embargo, se pasa de un contexto de análisis de tráfico de paquetes en redes a la comparación de similitud de conjuntos de secuencias genómicas, o genomas de manera abreviada. Esto quiere decir, que las variables que se consideran ahora, en este contexto, son el conjunto de las subsecuencias k-mer de largo k más frecuentes presentes en la distribución completa de frecuencias de algún genoma en particular (Se recomienda ver el glosario por la definición del término k-mer, y además se recomienda observar el ejemplo gráfico descriptivo del Anexo 2). No obstante, la estimación de entropía H de la combinación de las distribuciones de secuencias, denotada por M_{AB} (2.8) requiere definir ciertos aspectos. De aquí en adelante, se utiliza el término distribución para describir al conjunto de todas las frecuencias estimadas de secuencias k-mer distintas obtenidas de algún genoma en particular.

La estimación de entropía H de una distribución es análoga al método mostrado en la sección teórica. Sin embargo, la estimación de entropía de una distribución combinación $H(M_{AB})$ presente en la expresión (2.7) presenta diferencias con respecto a la estimación de entropía H de una única distribución. Intuitivamente, se puede entender por $H(M_{AB})$ como la entropía del promedio de las distribuciones A y B, es decir, la entropía resultante al promediar las frecuencias m_{Ai} y m_{Bi} de todos los elementos coincidentes en las distribuciones de frecuencias A y B. Sin embargo, en la fórmula de estimación de entropía H es equivalente el considerar su promedio o su suma. Esto se demuestra reemplazando los términos en la fórmula original del cálculo exacto de la Entropía de Shannon (2.1), lo cual queda expresado en (3.1). Nótese que la frecuencia de una secuencia k-mer m_C de un elemento i, $i \in \{1, \dots N\}$ equivale a promediar las frecuencias de dicha secuencia en las distribuciones A y B, esto es $\frac{m_{Ai}+m_{Bi}}{2}$ con $i \in \{1, \dots N\}$. Además, M_C denota al promedio del

conteo de las frecuencias totales de las secuencias presentes en las distribuciones A y B, esto es $\frac{(M_A+M_B)}{2}$.

$$H = \sum_{i=1}^{N}\left(\frac{m_C}{M_C} * log_2\left(\frac{m_C}{M_C}\right)\right) \leftrightarrow H = \sum_{i=1}^{N}\left(\frac{(m_{Ai}+m_{Bi})/_2}{(M_A+M_B)/_2} * log_2\left(\frac{(m_{Ai}+m_{Bi})/_2}{(M_A+M_B)/_2}\right)\right)$$

$$\leftrightarrow H = \sum_{i=1}^{N}\left(\frac{(m_{Ai}+m_{Bi})}{(M_A+M_B)} * log_2\left(\frac{(m_{Ai}+m_{Bi})}{(M_A+M_B)}\right)\right) \quad (3.1)$$

Entonces, es posible reemplazar en (2.6) las variables estimadas combinadas de las distribuciones A y B. Estas son, m_C que equivale a la suma de las frecuencias estimadas correspondientes a las secuencias k-mer coincidentes en las colas de prioridades de las dos distribuciones distintas A y B. Además, M_C equivale a la suma de los elementos totales M de ambas distribuciones A y B, L_C equivale a la suma acumulada de la frecuencia de las secuencias top-K coincidentes entre las distribuciones A y B, y, por último, N_C que equivale a la cardinalidad combinada estimada de ambas distribuciones (o cardinalidad unión estimada). Por lo tanto, la expresión resultante queda expresada en (3.2).

$$H(M_{AB}) = -\frac{1}{M_C}\left[\sum_{i=1}^{K} m_C * log\,(m_C) - L_C * log\,(M_C)\right] -$$

$$\frac{1}{M_C}\left[(M_C - L_C) * \left(\log(M_C - L_C) - log\big(M_C * (N_C - K)\big)\right)\right] \quad (3.2)$$

A continuación, se describen los algoritmos que permiten obtener la cardinalidad combinada (o unión) de dos distribuciones A y B, denotada por N_C, y la suma acumulada de la frecuencia de los elementos top-K coincidentes entre dos distribuciones A y B, denotada por L_C.

La estimación de la cardinalidad combinada N_C se realiza en base al cálculo de la media armónica Z, esto mediante el máximo valor entre los buckets de los vectores $A[i]$ y $B[i]$ del sketch del HyperLogLog (Algoritmo 3.1).

Algoritmo 3.1 Estimación de la unión de las cardinalidades N_A y N_B

Salida: Cardinalidad unión N_M

1: $\boldsymbol{para}\ i = 0\ to\ 2^{hll_p} - 1\ \boldsymbol{hacer}$
2: $\quad Z \leftarrow Z + 2^{-máx\,(A[i],B[i])}$
3: $\boldsymbol{fin\ para}$
4: $\boldsymbol{devolver}\ N_M \leftarrow\ \alpha_A * \frac{|A|^2}{Z}$

En segundo lugar, se calculan los términos del sumando del lado izquierdo de la expresión (3.2), estos son la suma acumulada de la frecuencia de los elementos top-K coincidentes entre dos distribuciones A y B denotada por L_C, el contador M_C, y la expresión sumatoria $left_C$ (3.3), según (Algoritmo 3.2), esto mediante la combinación de las colas de prioridad de ambas distribuciones de frecuencias.

$$left_C = \left(\sum_{i=1}^{K} m_C * log\,(m_C) \right) \tag{3.3}$$

El Algoritmo 3.2 describe el método de combinación de las colas de prioridad de las distribuciones A y B, para la estimación de los valores L_C y $left_C$. Además, se calcula el contador de secuencias totales M_C. Se utiliza una matriz que almacena variables de tipo $bool$ que registra aquellos elementos k-mer coincidentes de la cola de la distribución B. Posteriormente, se itera la misma matriz para considerar los elementos de B que no fueron considerados, y que deben ser sumados también a las variables L_C y $left_C$.

Algoritmo 3.2 Obtener término $left_C$ y frecuencia acumulada L_C de los top-k para $H(M_{AB})$

Entrada: Colas de prioridad Q_A y Q_B

Salida: Valores L_C y $left_C$

```
1:  M_C ← M_A + M_B
2:  L_C ← 0, left_C ← 0
3:  flagA[i][j] ← false, i ∈ {0, ..., 2^height}, j ∈ {0, ..., size(Q_B[i])}
4:  para address = 0 to 2^height − 1 hacer
5:      flag ← false
6:      para j = 0 to size(Q_A[address]) hacer
7:          para k = 0 to size(Q_B[address]) hacer
8:              si Q_A[address][j].first = Q_B[address][k].first entonces
9:                  m_c ← Q_A[address][j].second + Q_B[address][k].second
10:                 L_C ← L_C + m_C
11:                 left_C ← left_C + (m_C * log_2(m_C))
12:                 flagA[address][k] ← true
13:                 flag ← true
14:             fin si
15:         fin para
16:         si flag = false entonces
17:             m_C ← Q_A[address][j].second
18:             L_C ← L_C + m_C
19:             left_C ← left_C + (m_C * log_2(m_C))
20:         fin si
21:     fin para
22: fin para
23: para address = 0 to 2^height − 1 hacer
24:     para k = 0 to size(Q_B[address]) hacer
25:         si flagA[address][k] = false entonces
26:             m_C ← Q_B[address][k].second
27:             L_C ← L_C + m_C
28:             left_C ← left_C + (m_C * log_2(m_C))
29:         fin si
30:     fin para
31: fin para
```

Por lo tanto, se cuenta con N_C como la combinación de las cardinalidades N_A y N_B, también con L_C como la suma acumulada de la frecuencia de los elementos top-K coincidentes entre dos distribuciones A y B, el término $left_C$ (3.3), y el contador total M_C. Finalmente, se tienen las estimaciones de $H(M_{AB})$, $H(A)$ y $H(B)$ y, por lo tanto, es posible estimar la Divergencia de Jensen-Shannon (2.7). Luego, la estimación de la métrica Distancia Jensen-Shannon es directa, y equivale a la raíz cuadrada de la divergencia según la fórmula (2.9).

Capítulo 4

Experimentos y resultados

La sección de experimentación se divide en cinco partes. La primera parte describe brevemente las funciones hash utilizadas, y además se realiza un análisis de las colisiones en base al cálculo de entropía real y el valor de las variables involucradas en la expresión de estimación de H (2.6). En la segunda parte se muestran los análisis de resultados de la estimación de cardinalidad mediante el algoritmo HyperLogLog sketch, en base al Error relativo ($Error\ R.$). La tercera parte reúne los análisis de resultados de la estimación de entropía según distintas dimensiones del sketch del CountMin-CU y de la estructura de la cola de prioridad, en base a la Raíz del error cuadrático medio ($RMSE$) (4.1). Luego, en la cuarta parte se muestran los análisis de resultados de la estimación de la Distancia Jensen-Shannon en base al $RMSE$, esto de acuerdo con la selección de los mejores valores de los parámetros analizados en la segunda y tercera parte. Finalmente, se muestran gráficos que resumen el espacio y tiempo utilizado por los algoritmos de estimación en comparación al cálculo de la métrica exacta.

Los errores absolutos y relativos ayudan a tener una idea de cuán precisas son las estimaciones de cada ejecución en particular. Además, el RMSE describe cuán cerca en su conjunto están las estimaciones de los cálculos reales, y se calcula mediante (4.1), en base al error absoluto de la estimación.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Valor\ exacto - Valor\ estimado)^2}{n}} \tag{4.1}$$

Las características de los conjuntos de datos de genomas son:

- 20 conjuntos de datos de genomas distintos de la base de datos RefSeq de NCBi [8] (Tabla 4.1).
- Cantidad de secuencias de k-mer en el rango de 7 a 15 millones (Tabla 4.2).
- Secuencias compuestas por nucleótidos A C T G.
- Las estimaciones se realizaron en base a los 10-mer y 15-mer originales de los datasets (kmer no canónicos).

Estas longitudes de k-mer han sido utilizados, por ejemplo, en trabajos tales como [18], en la identificación de péptidos específicos de proteínas, o en [19] para una búsqueda optimad de la longitud k, en el contexto de la clasificación de una amplia gama de familias de virus conocidas.

Tabla 4.1: Notación de los conjuntos de genomas y cada identificador utilizado.

Genoma	ID	Genoma	ID
GCF_010894405.1_ASM1089440v1_genomic	10	GCF_000418325.1_ASM41832v1_genomic	41
GCF_001522075.1_ASM152207v1_genomic	12	GCF_001522455.1_ASM152245v1_genomic	45
GCF_000373685.1_ASM37368v1_genomic	14	GCF_900146605.1_329_genomic	46
GCF_000331305.1_ASM33130v1_genomic	15	GCF_900147295.1_403_genomic	47
GCF_001931535.1_ASM193153v1_genomic	19	GCF_001453855.1_WH-SGI-V-07256_genom	56
GCF_002950945.1_ASM295094v1_genomic	29	GCF_000629185.1_Pseu_aeru_3580_V1_geno	62
GCF_001533405.1_ASM153340v1_genomic	33	GCF_001450045.1_WH-SGI-V-07240_genom	72
GCF_900147355.1_410_genomic	35	GCF_001451065.1_WH-SGI-V-07392_genom	73
GCF_001527385.1_ASM152738v1_genomic	38	GCF_001531825.1_ASM153182v1_genomic	82
GCF_004010305.2_ASM401030v2_genomic	40	GCF_001978835.1_ASM197883v1_genomic	83

Tabla 4.2: Características del conjunto de genomas utilizados. Se muestra M que corresponde al número de secuencias del conjunto y la cantidad de bases totales. En el lado izquierdo se tabulan los dataset en el caso 10-mer, y al lado derecho el caso 15-mer.

Dataset	k	M	Dataset	k	M	Bases
10	10	13.434.814	10	15	13.434.809	13.434.823
12	10	7.225.550	12	15	7.225.545	7.225.559
14	10	8.808.850	14	15	8.808.845	8.808.859
15	10	11.584.384	15	15	11.584.379	11.584.393
19	10	16.040.657	19	15	16.040.652	16.040.666
29	10	14.557.580	29	15	14.557.575	14.557.589
33	10	7.115.412	33	15	7.115.407	7.115.421
35	10	7.451.298	35	15	7.451.293	7.451.307
38	10	8.249.648	38	15	8.249.643	8.249.657
40	10	13.473.535	40	15	13.473.530	13.473.544
41	10	14.782.116	41	15	14.782.111	14.782.125
45	10	7.649.851	45	15	7.649.846	7.649.860
46	10	6.885.083	46	15	6.885.078	6.885.092
47	10	7.161.141	47	15	7.161.136	7.161.150
56	10	7.235.005	56	15	7.235.000	7.235.014
62	10	6.918.757	62	15	6.918.752	6.918.766
72	10	7.485.504	72	15	7.485.499	7.485.513
73	10	7.021.143	73	15	7.021.138	7.021.152
82	10	7.460.467	82	15	7.460.462	7.460.476
83	10	7.407.494	83	15	7.407.489	7.407.503

4.1. Funciones hash y colisiones

La función hash utilizada en el valor de entrada de los sketches del HyperLogLog, CountMin-CU y la cola de prioridad, pertenece a la librería $bits/functional\_hash.h$ de C++, y está basada en $Murmurhash$ [20]. Sin embargo, el sketch perteneciente al algoritmo CountMin-CU utiliza otras $depth$ adicionales funciones $Murmurhash$ de 32 bits [21], las cuales son independientes por pares.

En la Tabla 4.3 y Tabla 4.4 se muestran los resultados del cálculo exacto de la entropía H_Real, en base al valor de cadena de texto del k-mer, y H_Real^* en base al valor hash del elemento. Además, se muestran los valores exactos de las variables involucradas en el método de estimación de entropía (2.6), y descritas en la sección 2.1. Por un lado, L es la suma acumulada de las frecuencias de los elementos top-K, en base a los valores de cadena de texto del k-mer, y, por otro lado, L^* es la misma suma acumulada de las frecuencias, pero en base al valor hash del elemento k-mer (hash de C++). Análogamente, N es la cardinalidad en base al valor de cadena de texto del k-mer, y N^* la misma en base al valor hash del elemento k-mer. De acuerdo con la comparación de las entropías H_Real y H_Real^*, se puede concluir que las funciones hash utilizadas presentan una muy baja probabilidad de colisión, esto debido a que el valor calculado de las variables L y N de las secuencias k-mer representadas en cadena de texto versus su valor representado en valor hash varían en muy baja magnitud, lo cual repercute de manera casi nula en la precisión del cálculo exacto de la entropía H y, en consecuencia, también de las estimaciones de las secciones posteriores. Por ejemplo, para el caso 10-mer, la suma acumulada L de las secuencias k-mer en cadena de texto se mantiene con el mismo valor en el caso en que se calcula mediante su valor hash. Lo mismo para la cardinalidad N, existe una muy baja diferencia en el resultado, lo que repercute ínfimamente en el cálculo de la entropía H. La cardinalidad N varía con una mayor magnitud en el caso 15-mer, sin embargo, es casi despreciable en el cálculo de la entropía H, al igual que en el caso 10-mer.

Tabla 4.3: Resultados de cálculo exacto de entropía real H_{Real}, en caso 10-mer. Se consideran 128 top-K, y se muestran la cantidad total de secuencias M, la suma acumulada de los valores top-K, y la cardinalidad N, correspondientes al cálculo según los valores de cadena de texto de las secuencias k-mer. Las demás variables L^*, N^* y H^*_{Real}, hacen referencia al cálculo de estas, en base al valor hash de la secuencia k-mer.

Dataset	k	top-k	M	L	L*	N	N*	H_Real	H_Real*
10	10	128	13.434.814	85.173	85.173	720.403	720.361	18,02	18,02
12	10	128	7.225.550	46.631	46.631	804.268	804.204	18,42	18,42
14	10	128	8.808.850	96.941	96.941	606.631	606.594	17,68	17,68
15	10	128	11.584.384	33.136	33.136	953.971	953.866	19,22	19,22
19	10	128	16.040.657	174.095	174.095	773.226	773.165	18,02	18,02
29	10	128	14.557.580	214.131	214.131	764.934	764.883	17,80	17,80
33	10	128	7.115.412	91.078	91.078	728.472	728.414	17,92	17,92
35	10	128	7.451.298	46.401	46.401	829.410	829.338	18,49	18,49
38	10	128	8.249.648	96.881	96.881	785.941	785.871	18,07	18,07
40	10	128	13.473.535	78.565	78.565	927.423	927.335	18,63	18,63

Tabla 4.4: Resultados de cálculo exacto de entropía real H_{Real}, en caso 15-mer. Las variables se definen de manera análoga a la Tabla 4.3.

Dataset	k	top-k	M	L	L*	N	N*	H_Real	H_Real*
10	15	128	13.434.809	7.952	7.952	11.426.941	11.411.742	23,33	23,32
12	15	128	7.225.545	2.088	2.089	6.688.485	6.683.445	22,62	22,62
14	15	128	8.808.845	7.288	7.288	7.542.159	7.535.550	22,73	22,73
15	15	128	11.584.379	13.440	13.441	10.893.541	10.879.776	23,32	23,32
19	15	128	16.040.652	7.434	7.434	13.256.022	13.235.394	23,51	23,50
29	15	128	14.557.575	8.625	8.626	11.611.764	11.596.068	23,27	23,27
33	15	128	7.115.407	2.877	2.877	6.223.562	6.219.073	22,46	22,46
35	15	128	7.451.293	2.004	2.005	6.934.517	6.929.015	22,68	22,68
38	15	128	8.249.643	3.048	3.048	7.211.233	7.205.149	22,68	22,67
40	15	128	13.473.530	6.806	6.806	11.830.020	11.813.799	23,41	23,40

4.2. Resultados de la estimación de cardinalidad mediante HLL sketch

Las pruebas de estimación de cardinalidad N_HLL de los conjuntos de genomas por medio del algoritmo HyperLogLog se realizaron en base a dos distintos largos de k-mer, estos son, 10-mer y 15-mer y a tres distintos valores del parámetro de precisión del sketch hll_p (denotado por HLL_P en la Tabla 4.5 y Tabla 4.6), estos son 11, 12 y 13, los cuales determinan la dimensión del vector de buckets del algoritmo. En la Tabla 4.5 y Tabla 4.6 se muestran los errores absolutos y relativos de estimación de cardinalidad, en ambos casos de k-mer, para un subconjunto de los datasets utilizados, estos son 12, 19 y 29.

Tabla 4.5: Resultados de estimación de cardinalidad N_{HLL}, en caso 10-mer. Se muestran el parámetro de precisión HLL_p, la cantidad total de secuencias M, y la cardinalidad exacta N_{Real} de cada dataset.

Dataset	k	HLL_P	M	N_HLL	N_Real	Error A.	Error R.
12	10	11	7.225.550	805.025	804.268	757	0,09%
12	10	12	7.225.550	803.092	804.268	1.176	0,15%
12	10	13	7.225.550	804.455	804.268	187	0,02%
19	10	11	16.040.657	748.285	773.226	24.941	3,23%
19	10	12	16.040.657	761.666	773.226	11.560	1,50%
19	10	13	16.040.657	767.855	773.226	5.371	0,69%
29	10	11	14.557.580	749.551	764.934	15.383	2,01%
29	10	12	14.557.580	752.321	764.934	12.613	1,65%
29	10	13	14.557.580	759.441	764.934	5.493	0,72%

Tabla 4.6: Resultados de estimación de cardinalidad N (N_{HLL}), en caso 15-mer. Se muestran el parámetro de precisión HLL_p, la cantidad total de secuencias M, y la cardinalidad exacta N_{Real} de cada dataset.

Dataset	k	HLL_P	M	N_HLL	N_Real	Error A.	Error R.
12	15	11	7.225.545	6.731.999	6.688.485	43.514	0,65%
12	15	12	7.225.545	6.605.200	6.688.485	83.285	1,25%
12	15	13	7.225.545	6.700.755	6.688.485	12.270	0,18%
19	15	11	16.040.652	13.215.517	13.256.022	40.505	0,31%
19	15	12	16.040.652	13.276.021	13.256.022	19.999	0,15%
19	15	13	16.040.652	13.158.836	13.256.022	97.186	0,73%
29	15	11	14.557.575	11.461.437	11.611.764	150.327	1,29%
29	15	12	14.557.575	11.296.429	11.611.764	315.335	2,72%
29	15	13	14.557.575	11.313.658	11.611.764	298.106	2,57%

A continuación, se muestran los gráficos de la Figura 4.1 y Figura 4.2, los cuales evidencian los errores relativos de estimación de cardinalidad para los casos 10 y 15-mer de todos los conjuntos de genomas.

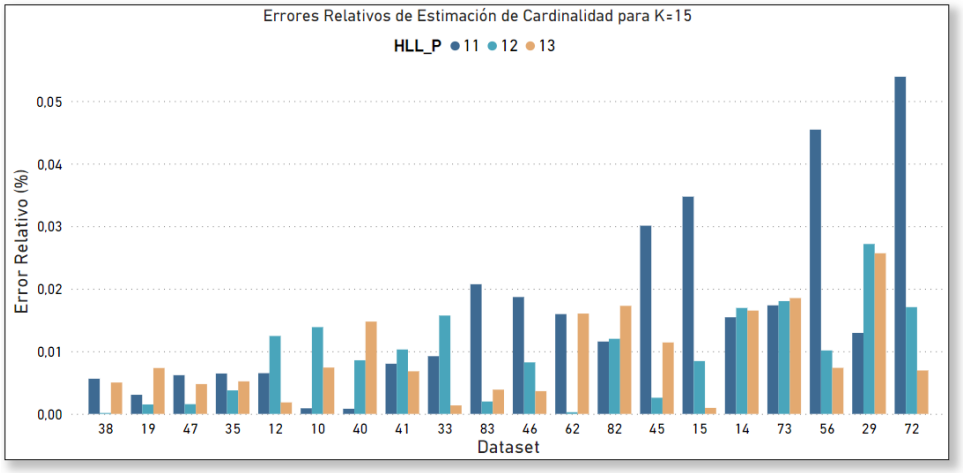


Figura 4.1: Errores relativos de estimación de cardinalidad N para caso 15-mer

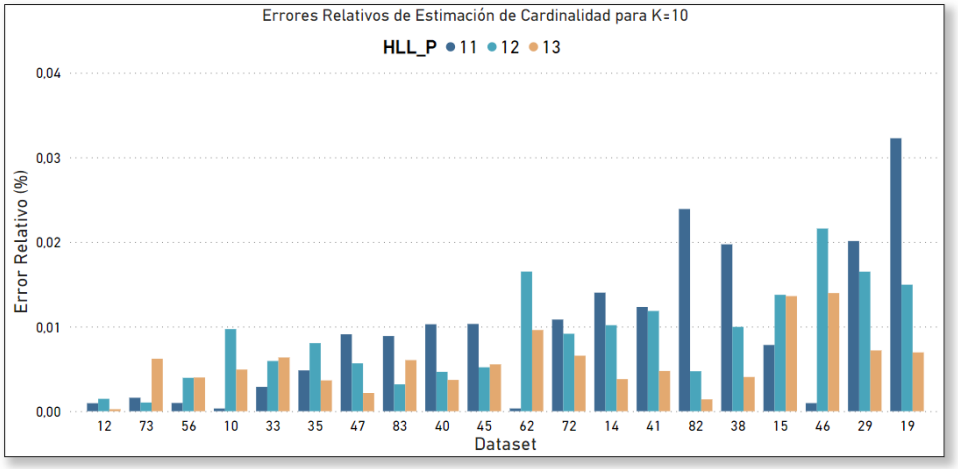


Figura 4.2: Errores relativos de estimación de cardinalidad N para caso 10-mer

Se puede observar que los errores relativos son inferiores al 4% en el caso 10-mer y al 6% en el caso 15-mer. Se considera que todas las estimaciones de cardinalidad entregan buenos resultados, ya que en ambos casos de k-mer se presentan errores relativos bajos y, por lo tanto, es factible utilizar cualquiera de los tres valores del parámetro de precisión para las estimaciones siguientes. En general, los resultados indican una muy buena estimación de cardinalidad y se utilizará el valor 13 del parámetro de precisión hll_p en las estimaciones de las secciones siguientes.

En el gráfico siguiente de la Figura 4.3, se muestra el $RMSE$ para cada valor del parámetro de precisión hll_p utilizado, y se evidencia que el tamaño del vector A está relacionado directamente al $RMSE$ de las estimaciones. EL $RMSE$ se calcula en base a los errores absolutos de estimación de la cardinalidad N de los conjuntos de genomas, mediante (4.1).

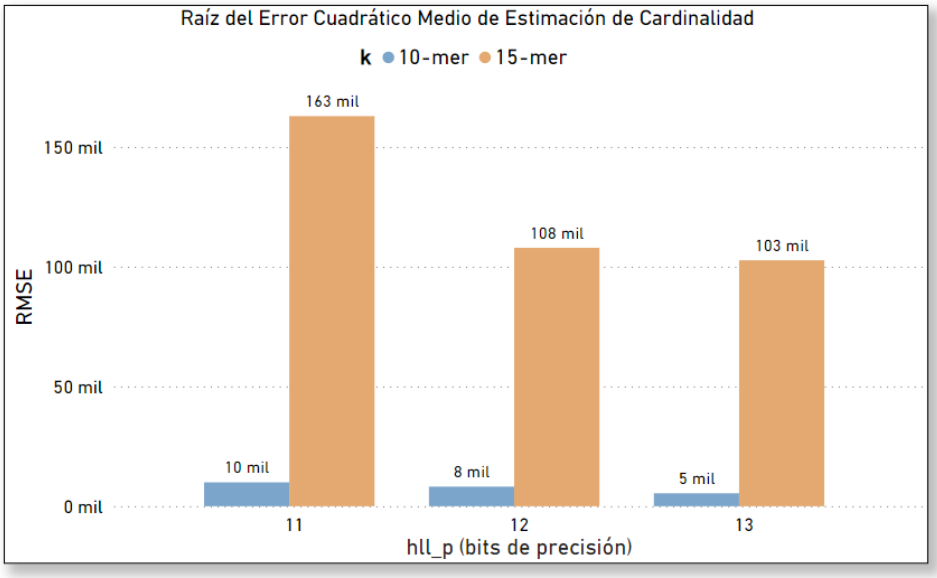


Figura 4.3: $RMSE$ de estimación de cardinalidad N, según parámetro hll_p, en caso 10 y 15-mer.

4.3. Resultados de estimación de Entropía de Shannon

Las pruebas de la estimación de la Entropía de Shannon H_est de los conjuntos se realizaron en base a dos distintos largos de k-mer, estos son, 10 y 15-mer; tres distintos tamaños de sketches de estimación de frecuencias del Countmin-CU, estos son, 4096, 8192 y 12288 buckets; y siete distintos tamaños de cola de prioridad de los elementos top-K, las cuales se limitan a 32, 64, 128, 256, 512, 1024 y 2048 elementos. Para el sketch del Countmin-CU su dimensión viene dada por $2^w \cdot d$ buckets, donde w (CM_{width}) es un parámetro de precisión que corresponde a la cantidad de bits de los valores hash que se utilizan para acceder a las direcciones de los contadores de frecuencia del elemento y d (CM_{depth}) es la profundidad que indica la cantidad de funciones hash utilizadas. Para la estructura de la cola de prioridad PQ su dimensión viene dada por $2^h \cdot 2^w$ elementos, donde h (PQ_{height}) es un parámetro de precisión que corresponde a la cantidad de bits del valor hash que se utilizan para acceder a las direcciones de las colas de prioridad de los elementos top-K, y w (PQ_{width}) es un parámetro que define el tamaño de cada una de estas colas.

En la Tabla 4.7 se muestran los parámetros utilizados que dimensionan cada una de las estructuras, la idea fue variar ambos en cada caso, de manera de tener una muestra más variada de estimaciones de la cual poder sacar conclusiones. Nótese que omito el valor del parámetro de precisión hll_p, el cual fue fijado en el valor 13 para las estimaciones posteriores de esta sección de experimentación.

Tabla 4.7: Valores de distintos tamaños de cola de prioridad, según los parámetros PQ_{height} y PQ_{width}, y distintos tamaños de sketch de estimación de frecuencias del algoritmo CountMin-CU, según los parámetros de CM_{width} y CM_{depth}.

top-k	PQ_Height	PQ_Width
32	3	2
64	4	2
128	4	3
256	4	4
512	5	4
1024	6	4
2048	6	5
CMSketch	**CM_Width**	**CM_Depth**
4096	11	2
8192	12	2
12288	12	3

A continuación, en la Tabla 4.8 y Tabla 4.9 se muestran algunos resultados, junto a los errores absolutos y relativos de estimación de entropía para los casos 10 y 15-mer, respectivamente. Se debe aclarar que L_{PQ} es la suma acumulada de las frecuencias de los k-mers considerados top-K, y L_{Real} es la suma acumulada exacta de las frecuencias reales.

Los grupos de columnas separados por color representan, por un lado, los resultados de estimación de entropía H (color anaranjado) versus los valores exactos (color verde). Además, el valor de la entropía es separado en sus dos términos $left$ y $right$, los que evidencian el aporte de los elementos más frecuentes $(left)$ y el aporte de la distribución uniforme $(right)$ de los elementos restantes considerados menos frecuentes. Los valores de estas variables son negativas ya que su cálculo se corresponde con las fórmulas (2.4) y (2.5), y es debido a que la evaluación de $log_2(x)$ resulta en valores negativos para valores de $x \in]0, \dots, 1[$.

Tabla 4.8: Estimaciones de entropía H sobre dataset 12, en caso 10-mer. Los valores de parámetros del CountMin-CU sketch y la cola de prioridad se variaron de acuerdo con la Tabla 4.7. Se muestran errores absolutos (ErrorA.), errores relativos (ErrorR.) y tiempos de ejecución en segundos T(seg).

Dataset	k	top-k	CMSketch	L_PQ	L_Real	H_Est	H_Est_n	Left	Right	H_Real	H_Real_n	Left_R	Right_R	ErrorA.	ErrorR.	T (seg)
12	10	256	4.096	555.307	83.502	19,11	0,97	-0,90	-18,21	18,42	0,94	-0,17	-18,26	0,035	3,7%	7,4
12	10	256	8.192	275.254	83.502	19,41	0,99	-0,48	-18,92	18,42	0,94	-0,17	-18,26	0,050	5,3%	7,4
12	10	256	12.288	227.120	83.502	19,45	0,99	-0,41	-19,04	18,42	0,94	-0,17	-18,26	0,052	5,6%	8,0
12	10	512	4.096	1.110.499	146.204	18,60	0,95	-1,80	-16,80	18,42	0,94	-0,30	-18,13	0,009	1,0%	7,5
12	10	512	8.192	550.410	146.204	19,19	0,98	-0,97	-18,23	18,42	0,94	-0,30	-18,13	0,039	4,2%	7,5
12	10	512	12.288	454.183	146.204	19,29	0,98	-0,82	-18,47	18,42	0,94	-0,30	-18,13	0,044	4,7%	8,0
12	10	1024	4.096	2.220.529	248.285	17,55	0,89	-3,60	-13,95	18,42	0,94	-0,51	-17,92	0,045	4,8%	14,4
12	10	1024	8.192	1.100.372	248.285	18,76	0,96	-1,94	-16,83	18,42	0,94	-0,51	-17,92	0,017	1,8%	14,2
12	10	1024	12.288	907.952	248.285	18,95	0,97	-1,63	-17,32	18,42	0,94	-0,51	-17,92	0,027	2,9%	14,8

Tabla 4.9: Estimaciones de entropía H sobre dataset 12, en caso 15-mer. Los valores de parámetros del CountMin-CU sketch y la cola de prioridad se variaron de acuerdo con la Tabla 4.7. Se muestran errores absolutos (ErrorA.), errores relativos (ErrorR.) y tiempos de ejecución en segundos T(seg).

Dataset	k	top-k	CMSketch	L_PQ	L_Real	H_Est	H_Est_n	Left	Right	H_Real	H_Real_n	Left_R	Right_R	ErrorA.	ErrorR.	T (seg)
12	15	256	4.096	561.108	3.797	21,91	0,97	-0,91	-21,00	22,62	1,00	-0,01	-22,61	0,031	3,1%	7,8
12	15	256	8.192	280.839	3.797	22,32	0,99	-0,49	-21,83	22,62	1,00	-0,01	-22,61	0,012	1,3%	7,7
12	15	256	12.288	231.414	3.797	22,39	0,99	-0,42	-21,98	22,62	1,00	-0,01	-22,61	0,009	1,0%	8,2
12	15	512	4.096	1.122.014	6.278	21,16	0,93	-1,81	-19,34	22,62	1,00	-0,02	-22,60	0,064	6,5%	7,6
12	15	512	8.192	561.553	6.278	21,99	0,97	-0,99	-21,00	22,62	1,00	-0,02	-22,60	0,027	2,8%	8,2
12	15	512	12.288	462.617	6.278	22,12	0,98	-0,83	-21,29	22,62	1,00	-0,02	-22,60	0,021	2,2%	8,9
12	15	1024	4.096	2.243.806	10.274	19,62	0,87	-3,63	-15,99	22,62	1,00	-0,03	-22,59	0,132	13,3%	15,2
12	15	1024	8.192	1.122.716	10.274	21,31	0,94	-1,97	-19,34	22,62	1,00	-0,03	-22,59	0,057	5,8%	14,3
12	15	1024	12.288	924.953	10.274	21,59	0,95	-1,66	-19,93	22,62	1,00	-0,03	-22,59	0,045	4,6%	15,1

En primer lugar, es evidente que el valor de estimación de la frecuencia de los elementos top-K, es decir, L_{PQ} está muy sobreestimado en comparación con el cálculo exacto. Sin embargo, el valor de esta variable disminuye a medida que se aumenta la dimensión del sketch del CountMin-CU, esto es, al aumentar el parámetro de precisión $width$ y la cantidad $depth$ de funciones hash utilizadas. Dicho esto, y considerando que es una variable importante en la fórmula de estimación de la entropía, su sobreestimación conlleva a resultados, en general, muy positivos. Esto es así debido a que en el método de estimación se considera el termino derecho $right$ como la entropía de una distribución uniforme que también es dependiente del valor de la variable L_{PQ}, es decir, aunque se sobreestime el resultado de entropía de los elementos más frecuentes, el aporte de la distribución uniforme al valor de la entropía puede balancear el valor final de la estimación. Además, se

debe hacer notar que un valor L_{PQ} menos sobreestimado no necesariamente conlleva a una mejor estimación de entropía.

Los siguientes gráficos de la Figura 4.4, Figura 4.5, y Figura 4.6 resumen los resultados obtenidos según el cálculo del $RMSE$, en los casos 10 y 15-mer, según los distintos tamaños de sketch del CM-CU y los distintos tamaños de la cola de prioridad de elementos top-K de la Tabla 4.7. El $RMSE$ se calcula en base a los errores absolutos de estimación de entropía normalizada H_n (2.2) de los conjuntos de genomas, mediante (4.1).

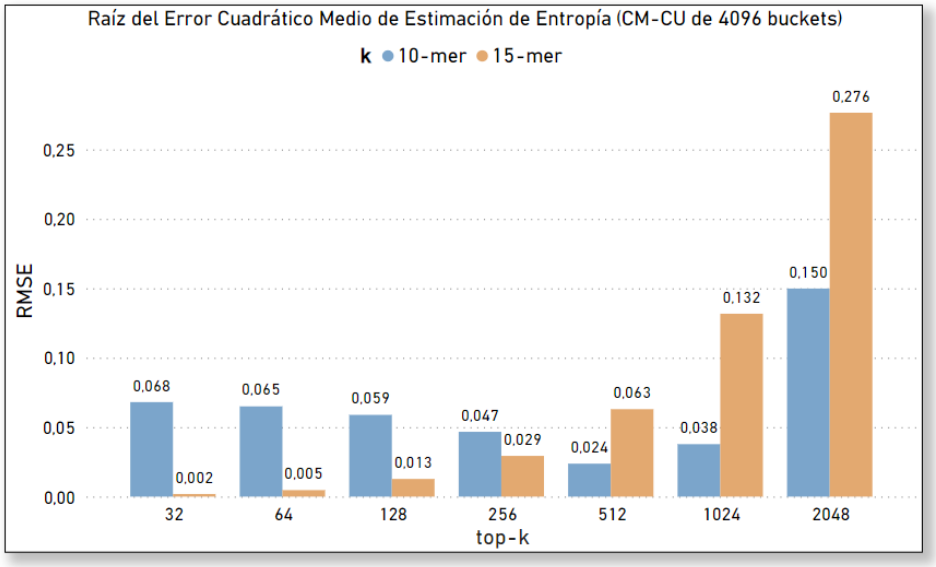


Figura 4.4: $RMSE$ de estimación de entropía H, en caso 10 y 15-mer, dado un tamaño de sketch de CountMin-CU de 4.096 buckets. Los tamaños de cola de prioridad admiten entre 32 y 2048 secuencias k-mer.

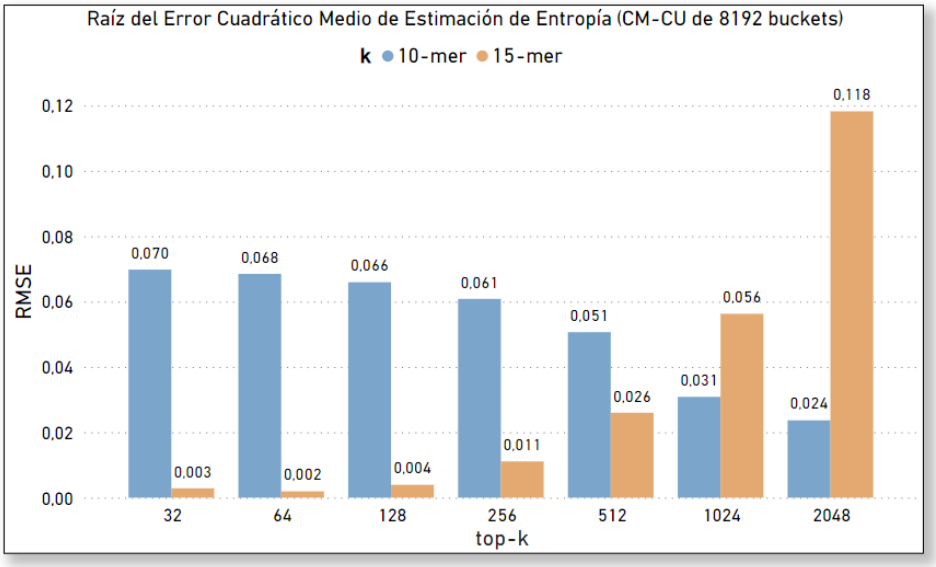


Figura 4.5: $RMSE$ de estimación de entropía H, en caso 10 y 15-mer, dado un tamaño de sketch de CountMin-CU de 8.192 buckets. Los tamaños de cola de prioridad admiten entre 32 y 2048 secuencias k-mer.

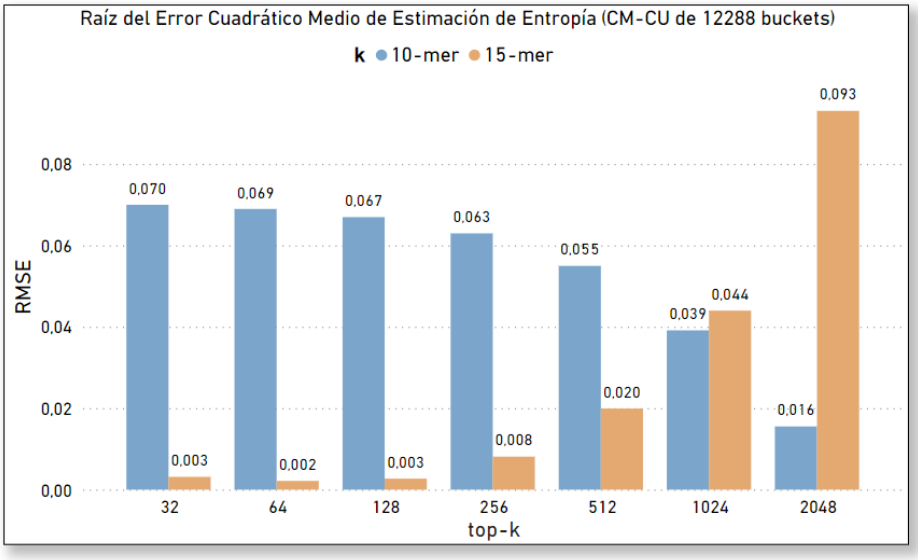


Figura 4.6: $RMSE$ de estimación de entropía H, en caso 10 y 15-mer, dado un tamaño de sketch de CountMin-CU de 12.288 buckets. Los tamaños de cola de prioridad admiten entre 32 y 2048 secuencias k-mer.

En primer lugar, y de acuerdo con las tres figuras anteriores, en el caso 15-mer se puede observar que el $RMSE$ más bajo se obtiene a menor cantidad de elementos top-K considerados, es decir, a menores dimensiones de la cola de prioridad. Por otra parte, para el caso 10-mer, el $RMSE$ más bajo, en cambio, se obtiene a mayores dimensiones de la cola de prioridad. Sin embargo, según la Figura 4.4, se da una situación particular para las dimensiones de cola de prioridad mayores a 512 elementos. Esto se explica debido a que la dimensión del sketch de estimación de frecuencias no es la adecuada para esa cantidad de elementos top-K considerados, es decir, está subdimensionada, y 11 bits de precisión para el parámetro $width$ conlleva a un valor de L_{PQ} muy sobreestimado.

En la Figura 4.5 y Figura 4.6, la tendencia del $RMSE$ a la baja es constante a lo largo de todos los tamaños de cola de prioridad, es decir, a mayores dimensiones del sketch de estimación de frecuencias del algoritmo CountMin-CU, los resultados son mejores. Análogamente, en el caso 15-mer, a mayor dimensión del sketch del CounMin-CU los $RMSE$ son más bajos.

Estas conclusiones se pueden complementar analizando algunos histogramas de ejemplo de las distribuciones de frecuencias en los casos 10-mer y 15-mer, respectivamente, para los dataset 12 y 19, en la Figura 4.7, Figura 4.8, Figura 4.9 y Figura 4.10.

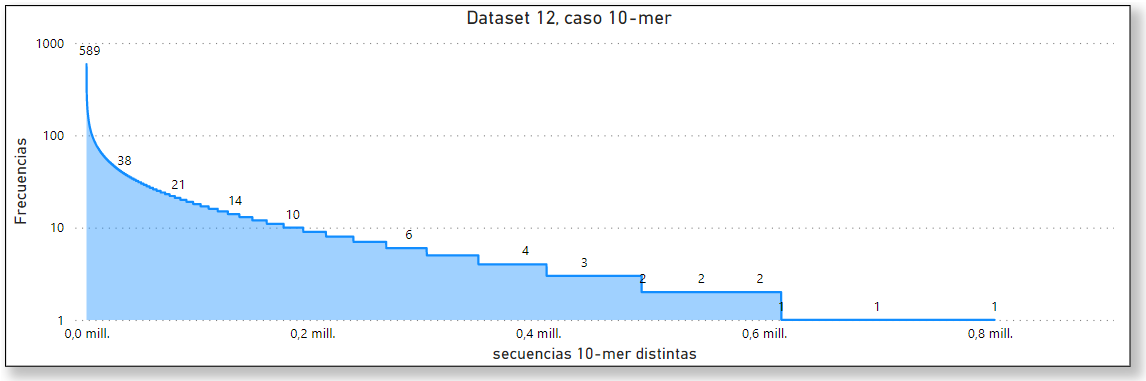


Figura 4.7: Histograma de distribución de frecuencias de las secuencias 10-mer del dataset 12.

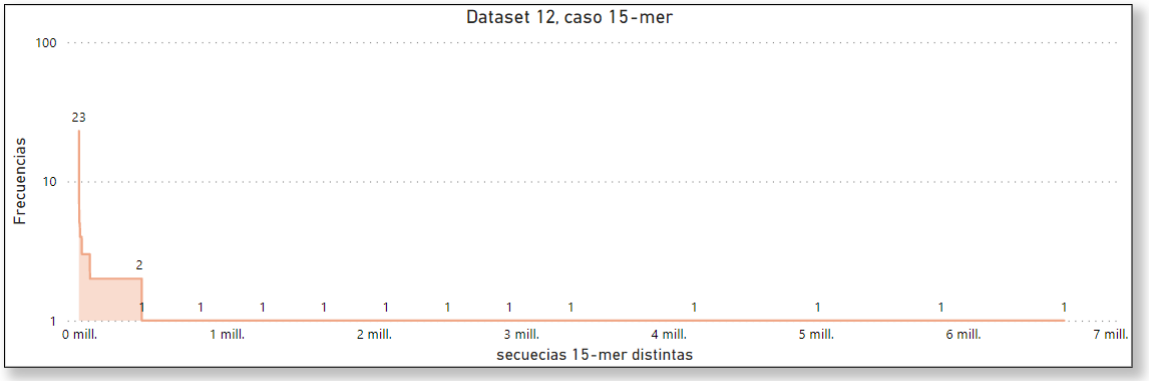


Figura 4.8: Histograma de distribución de frecuencias de las secuencias 15-mer del dataset 12.

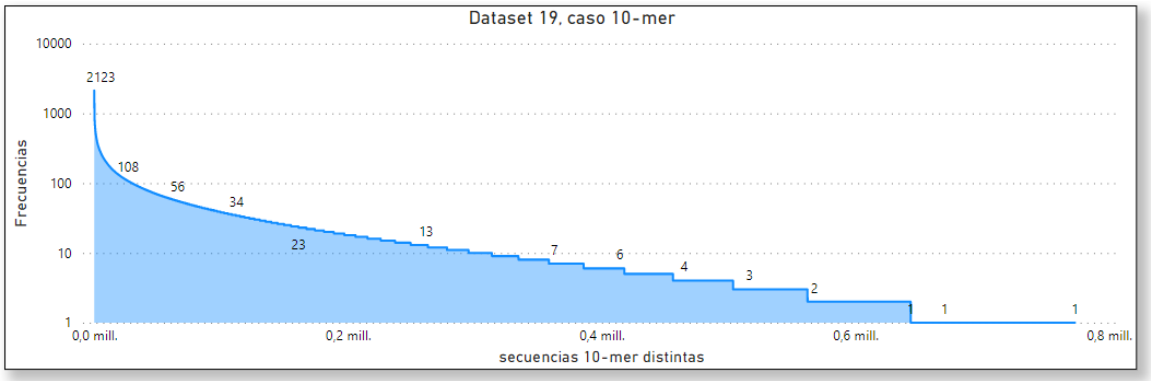


Figura 4.9: Histograma de distribución de frecuencias de las secuencias 10-mer del dataset 19.

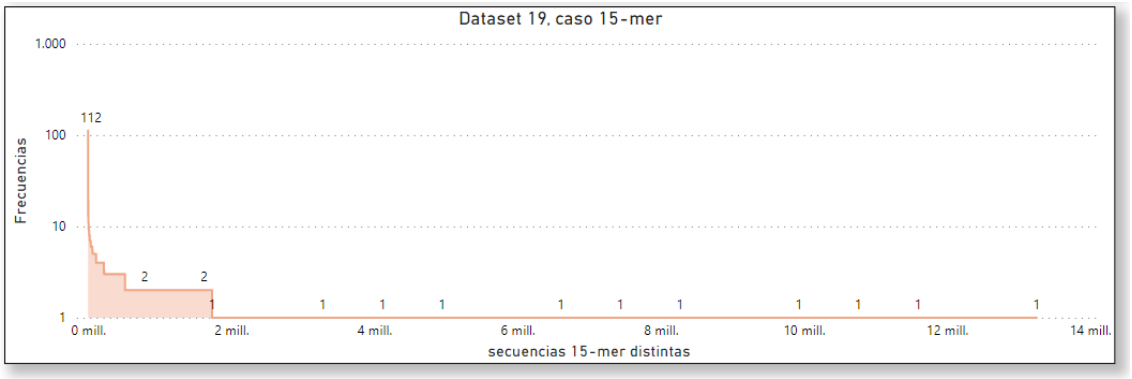


Figura 4.10: Histograma de distribución de frecuencias de las secuencias 15-mer del dataset 19.

Al comparar las distribuciones de los dataset 12 y 19, en los casos 10 y 15-mer, las diferencias son claras. En primer lugar, los valores de las frecuencias de los elementos top-K son menores en el caso 15-mer, lo que es lógico ya que a mayor largo de k-mer es menor la cardinalidad de cada elemento en particular. Además, se puede notar la diferencia en el lado derecho de las distribuciones en cada caso de k-mer, siendo mucho más uniforme el caso 15-mer, esto debido a que la cantidad de elementos repetidos es mucho más baja en comparación al caso 10-mer. En definitiva, de los histogramas podemos concluir que una buena estimación de entropía depende, primero, de la cantidad de elementos top-K considerados, y segundo, el cuan uniforme sea la distribución de los elementos del lado derecho de los histogramas.

A continuación, en la Figura 4.11 se muestra el $RMSE$ de estimación de entropía en base al conteo exacto de las frecuencias de los elementos top-K sin variar los demás parámetros, es decir, reemplazando únicamente las frecuencias estimadas por su frecuencia real.

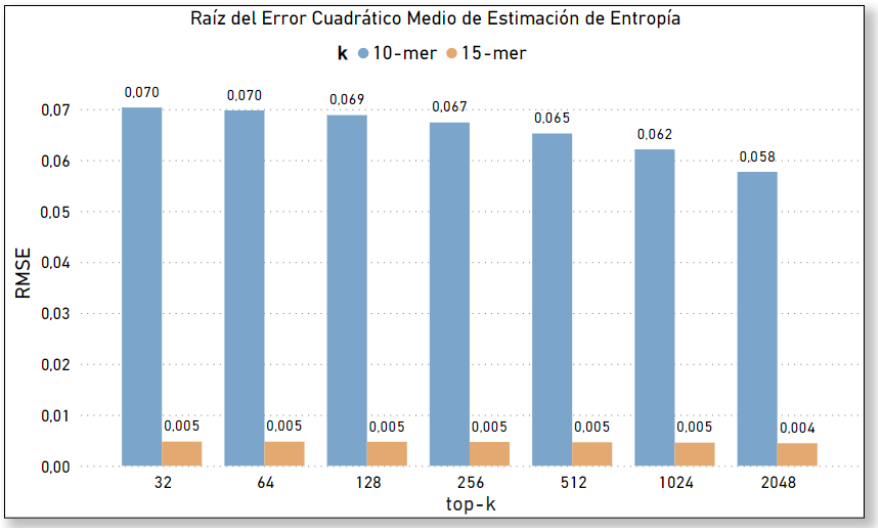


Figura 4.11: $RMSE$ de estimación de entropía H en base a conteo exacto de las secuencias k-mer más frecuentes o top-K.

Al comparar estos resultados con los de los gráficos de la Figura 4.4, 4.5 y 4.6, en los que se utiliza la frecuencia estimada de las secuencias top-K, se puede observar que para el caso 10-mer existen $RMSE$ muy similares, y la tendencia a la baja existe. Sin embargo, esta tendencia es mucho menor en la Figura 4.11, esto se debe a que la sobreestimación de la variable L_{PQ} permite obtener mejores resultados, pero limitando la cantidad de secuencias top-K consideradas. En el caso 15-mer la tendencia a la baja del $RMSE$ a menores tamaños de cola existe, pero es mucho menos notoria, si se compara con los resultados de entropía H utilizando el sketch de estimación de frecuencias. En efecto, se puede concluir que el uso del CountMin-CU es factible sólo si la cantidad de secuencias top-K consideradas en la estimación, se limita adecuadamente, y de acuerdo con el largo de k-mer. Se debe mencionar que no me fue posible realizar un análisis comparativo del error del procesamiento de las secuencias top-K mediante la cola de prioridad, debido a la complejidad en el tiempo de ejecución de la estructura clásica (STL). La principal ventaja de utilizar la estructura de cola basada en sketch es que gracias a los beneficios del hashing es mucho más rápido realizar el ordenamiento de frecuencias de las secuencias sobre colas de prioridad más pequeñas que de una sola gran cola de prioridad, por lo cual se permite disminuir radicalmente el tiempo de ordenamiento.

A continuación, y para finalizar esta parte de la experimentación, en la Tabla 4.10 se muestra la selección de valores de los parámetros de estimación de entropía obtenidos de los análisis anteriores, los cuales se utilizan en la estimación de la Distancia Jensen-Shannon de la sección siguiente. Además, en la Figura 4.12 se muestran los $RMSE$ de los resultados de estimación de entropía en base a esta elección de valores de parámetros.

Tabla 4.10: Mejores valores de parámetros de dimensión de CountMin-CU sketch, HLL sketch y `cola`, en casos 10 y 15-mer.

k	top-k	PQ_height	PQ_width	CM_width	CM_depth	HLL_P
10	512	5	4	12	3	13
10	1024	5	5	12	3	13
10	2048	6	5	12	3	13
15	64	4	2	12	3	13
15	128	4	3	12	3	13
15	256	4	4	12	3	13

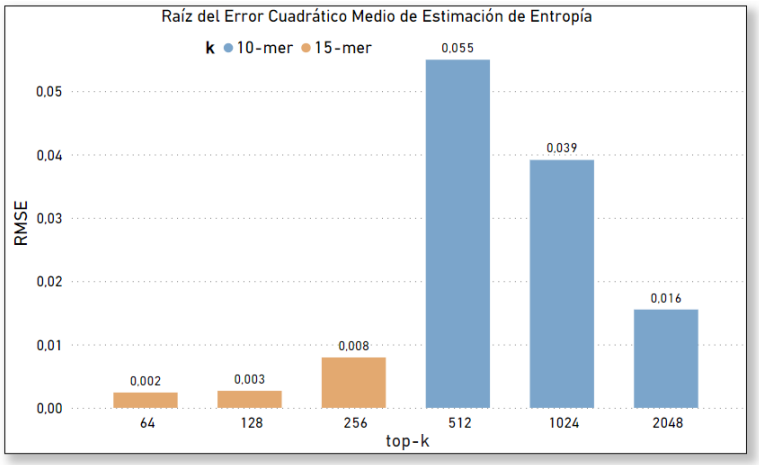


Figura 4.12: $RMSE$ de estimación de entropía H en base mejores valores de parámetros.

4.4. Resultados de la estimación de la Distancia Jensen-Shannon

En la Tabla 4.11 y Tabla 4.12 se muestran los resultados de las ejecuciones de estimación de la Distancia Jensen-Shannon para los casos 10-mer y 15-mer, en base a los tamaños de cola de prioridad de 128 y 1024 elementos. Se muestran el error absoluto de estimación, además de los tiempos de ejecución de estimación y de cálculo exacto de la métrica. Las variables N_C (N_c en la tabla) y L_C (L_c en la tabla), equivalen a la cardinalidad combinada entre los dataset y a la suma acumulada de las frecuencias de los top-K coincidentes entre ambos dataset, respectivamente. Además, M_1 y M_2 corresponden a la cantidad de secuencias k-mer totales de cada uno de los dataset (o distribuciones).

Tabla 4.11: Resultados de estimación de Distancia Jensen-Shannon en caso 10-mer. Se muestra la métrica estimada $JSDist$, la métrica exacta $JSDist.R$, junto con su error absoluto ($Error\ A$, y los tiempos de ejecución de la estimación $T(seg)$ y el cálculo exacto de la métrica $T(seg)R$. Los valores de los parámetros del CountMin-CU y la cola prioridad se establecen de acuerdo con la Tabla 4.10.

Dataset	k	top-k	M1	M2	N_c	L_c	JSDist.	JSDist.R	Error A.	T (seg)	T (seg) R.
10 | 83	10	1024	13.434.814	7.407.494	866.721	2.609.308	0,59	0,51	0,08	29,38	12,4
12 | 33	10	1024	7.225.550	7.115.412	889.470	1.791.069	0,56	0,42	0,14	18,56	8,8
14 | 15	10	1024	8.808.850	11.584.384	1.042.609	2.565.854	0,71	0,91	0,20	45,17	10,3
19 | 10	10	1024	16.040.657	13.434.814	843.854	3.667.332	0,54	0,37	0,17	39,83	14,7
19 | 29	10	1024	16.040.657	14.557.580	858.160	3.794.535	0,52	0,27	0,26	38,58	14,6
19 | 40	10	1024	16.040.657	13.473.535	963.512	3.686.899	0,54	0,35	0,19	38,03	15,0
19 | 62	10	1024	16.040.657	6.918.757	881.189	2.860.137	0,54	0,30	0,24	30,82	11,3
29 | 10	10	1024	14.557.580	13.434.814	840.731	3.480.053	0,54	0,34	0,20	37,5	13,7
29 | 40	10	1024	14.557.580	13.473.535	967.722	3.499.620	0,55	0,40	0,15	38,93	13,5
29 | 56	10	1024	14.557.580	7.235.005	895.375	2.712.509	0,55	0,24	0,31	30,09	11,4
35 | 15	10	1024	7.451.298	11.584.384	1.044.449	2.410.813	0,57	0,78	0,21	43,19	10,5
35 | 38	10	1024	7.451.298	8.249.648	922.982	1.962.253	0,55	0,39	0,16	21,71	9,0
40 | 10	10	1024	13.473.535	13.434.814	963.098	3.372.417	0,58	0,46	0,12	37,44	13,5
40 | 38	10	1024	13.473.535	8.249.648	980.025	2.721.070	0,54	0,43	0,11	27,24	11,5
41 | 10	10	1024	14.782.116	13.434.814	845.863	3.509.855	0,55	0,35	0,21	35,58	13,7
41 | 19	10	1024	14.782.116	16.040.657	855.274	3.824.337	0,52	0,27	0,26	38,22	14,7
41 | 29	10	1024	14.782.116	14.557.580	859.258	3.637.058	0,51	0,16	0,35	36,7	13,7
41 | 40	10	1024	14.782.116	13.473.535	966.234	3.529.422	0,55	0,40	0,16	37,58	13,9
45 | 46	10	1024	7.649.851	6.885.083	885.173	1.814.068	0,57	0,40	0,17	18,23	8,2
56 | 62	10	1024	7.235.005	6.918.757	866.860	1.778.111	0,50	0,19	0,32	16,81	8,2
72 | 73	10	1024	7.485.504	7.021.143	871.757	1.821.287	0,50	0,19	0,31	18,89	8,2
82 | 83	10	1024	7.460.467	7.407.494	867.644	1.859.528	0,55	0,25	0,29	19,16	7,9

De acuerdo con la Tabla 4.11, la estimación de la métrica en el caso 10-mer es aceptable. Sin embargo, en algunos pares de dataset el error absoluto aumenta drásticamente. Por otra parte, la diferencia en los tiempos de ejecución de estimación de la métrica versus el cálculo exacto es notoria, y se explica por el costo de ejecución del proceso de ordenamiento de los elementos top-K en la cola de prioridad. En consecuencia,

mientras más elementos top-K sean considerados en la estimación, el tiempo de ejecución también aumentará.

Tabla 4.12: Resultados de estimación de Distancia Jensen-Shannon en caso 15-mer. Se muestra la métrica estimada $JSDist$, la métrica exacta $JSDist.R$, junto con su error absoluto ($Error\ A$, y los tiempos de ejecución de la estimación $T(seg)$ y el cálculo exacto de la métrica $T(seg)R$. Los valores de los parámetros del CountMin-CU y la cola prioridad se establecen de acuerdo con la Tabla 4.10.

Dataset	k	top-k	M1	M2	N_c	L_c	JSDist.	JSDist.R	Error A.	T (seg)	T (seg) R.
10 | 83	15	128	13.434.809	7.407.489	17.003.179	334.074	1,01	0,98	0,023	11,7	40,5
12 | 33	15	128	7.225.545	7.115.407	12.266.604	229.632	0,97	0,96	0,019	8,5	31,9
14 | 15	15	128	8.808.845	11.584.379	18.827.539	332.060	1,02	1,01	0,006	11,9	59,9
19 | 10	15	128	16.040.652	13.434.809	22.681.261	472.022	0,94	0,92	0,026	17,3	65,8
19 | 29	15	128	16.040.652	14.557.575	22.861.023	489.079	0,95	0,88	0,074	18,7	76,4
19 | 40	15	128	16.040.652	13.473.530	23.652.768	472.082	0,96	0,94	0,025	17,2	83,2
19 | 62	15	128	16.040.652	6.918.752	18.805.100	368.333	1,01	0,99	0,020	13,2	63,2
29 | 10	15	128	14.557.575	13.434.809	21.437.651	448.059	0,96	0,91	0,052	18,8	65,9
29 | 40	15	128	14.557.575	13.473.530	21.409.612	448.119	0,95	0,94	0,011	17,5	58,6
29 | 56	15	128	14.557.575	7.235.000	17.014.992	348.424	0,98	0,96	0,017	12,9	51,4
35 | 15	15	128	7.451.293	11.584.379	17.735.202	308.373	1,01	1,02	0,003	11,0	74,8
35 | 38	15	128	7.451.293	8.249.643	13.417.249	251.380	0,96	0,95	0,009	9,6	42,3
40 | 10	15	128	13.473.530	13.434.809	22.133.843	431.062	0,97	0,95	0,021	15,7	62,6
40 | 38	15	128	13.473.530	8.249.643	18.011.508	347.612	0,98	0,97	0,018	12,2	51,8
41 | 10	15	128	14.782.111	13.434.809	21.283.641	452.177	0,94	0,91	0,034	17,4	55,1
41 | 19	15	128	14.782.111	16.040.652	22.751.593	493.197	0,93	0,87	0,062	17,5	70,2
41 | 29	15	128	14.782.111	14.557.575	19.529.691	469.234	0,88	0,80	0,079	17,0	65,2
41 | 40	15	128	14.782.111	13.473.530	21.574.589	452.237	0,94	0,93	0,005	17,7	67,4
45 | 46	15	128	7.649.846	6.885.078	12.446.176	232.751	0,97	0,95	0,013	9,1	36,5
56 | 62	15	128	7.235.000	6.918.752	9.606.994	227.678	0,75	0,67	0,075	8,4	36,3
72 | 73	15	128	7.485.499	7.021.138	9.895.538	232.206	0,76	0,68	0,075	8,7	36,9
82 | 83	15	128	7.460.462	7.407.489	11.790.868	238.000	0,93	0,88	0,049	8,5	37,7

De acuerdo con la Tabla 4.12, la estimación de la métrica en el caso 15-mer es mucho más precisa que en el caso 10-mer. Además, se pueden observar tiempos de ejecución de estimación que son favorables para el método, a diferencia del caso 10-mer.

En la Figura 4.13 se muestra el RMSE de estimación de Distancia Jensen-Shannon en base a los valores de los parámetros de la Tabla 4.10. Al igual que en las estimaciones de los puntos anteriores, el $RMSE$ se calcula en base a los errores absolutos de estimación de la métrica, según la fórmula (4.1).

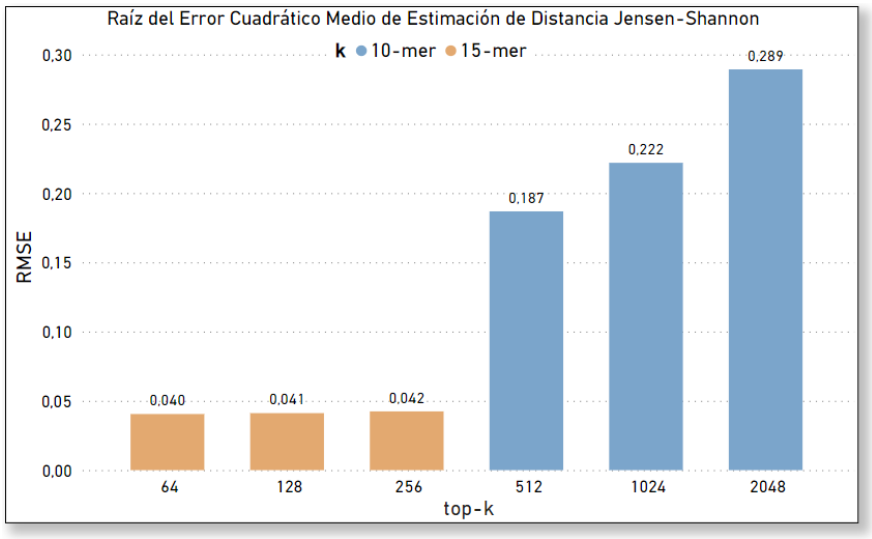


Figura 4.13: $RMSE$ de estimación de Distancia Jensen-Shannon en base a los parámetros de Tabla 4.10, de acuerdo con los casos 10 y 15-mer.

De acuerdo con la Figura 4.13, el $RMSE$ de estimación de la métrica es mucho menor en el caso 15-mer, en base a todos los tamaños de cola de prioridad seleccionados. Además, esto se condice con el análisis de los resultados de la estimación de entropía H de la sección 4.3, es decir, a menor cantidad de elementos top-K considerados, es menor el error de estimación. Sin embargo, en el caso 10-mer no resulta de la misma forma, puesto que, según el análisis de los resultados de estimación de entropía H de la sección anterior, se debió notar una tendencia a la baja del $RMSE$ a medida que se aumenta la cantidad de elementos top-K considerados, pero ocurre lo contrario. En efecto, esto sólo se puede explicar por la estimación de la entropía de la combinación $H(M_{AB})$, y a causa de la sobreestimación de la suma acumulada de las frecuencias combinadas de los top-K (L_C).

En la Figura 4.14 y Figura 4.15 se muestra el error absoluto de estimación de Distancia Jensen-Shannon sobre los pares de dataset utilizados en las pruebas de experimentación. En el caso 10-mer, se muestran errores absolutos en base a los tamaños de cola de prioridad de 512, 1024 y 2048 secuencias kmer. Por último, en el caso 15-mer, se muestran errores absolutos en base a los tamaños de cola de prioridad de 64, 128 y 256 secuencias.

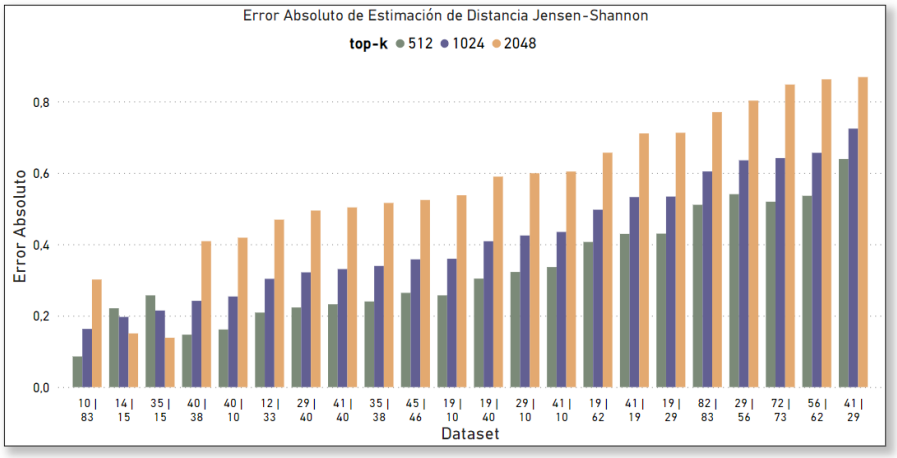


Figura 4.14: Errores absolutos de estimación de Distancia Jensen-Shannon en caso 10-mer, según valores de parámetros de Tabla 4.10.

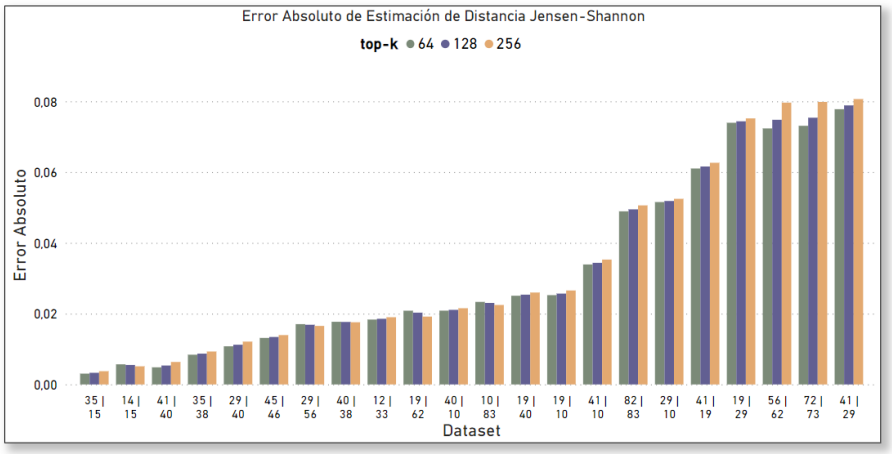


Figura 4.15: Errores absolutos de estimación de Distancia Jensen-Shannon en caso 15-mer, según valores de parámetros de Tabla 4.10.

4.5. Espacio utilizado y tiempos de ejecución promedio

Las características de hardware del equipo sobre el cual se ejecutaron las estimaciones son las siguientes:

- HP Envy 13 Laptop
- Procesador: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- RAM: 8,00 GB (7,75 GB utilizable)
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64
- S.O: Windows 11 Home Single Language

En la siguiente sección se muestra un resumen del espacio utilizado en las estimaciones, además de los promedios de los tiempos de ejecución de estimación de la métrica.

Primero, el espacio del HyperLogLog sketch depende del valor del parámetro de precisión hll_p, esto es 2^{hll_p} buckets, el cual almacena valores de enteros sin signo de 8 bits $uint8_t$ (1 byte), por lo tanto, el espacio utilizado es 2^{hll_p} bytes. Luego, para el Countmin-CU sketch se utiliza una matriz de $(depth * 2^{width})$ buckets, con $depth$ cantidad de funciones hash y $width$ como parámetro de precisión del largo de cada fila. Los buckets del sketch almacenan un dato entero de 32 bits $uint32_t$ (4 bytes), por lo tanto, el espacio utilizado por este es $(depth * 2^{width} * 4)$ bytes. Por último, la matriz de la cola de prioridad está compuesta por 2^{height} filas, donde $height$ es el parámetro de precisión de la cantidad de vectores de la matriz, y en donde cada uno de estos vectores puede almacenar un máximo de 2^{width} buckets de pares, donde $width$ es el parámetro que dimensiona el tamaño del vector. El par almacenable es del tipo $pair < uint32_t, int >$, por lo tanto, el espacio utilizado por la estructura es de $\left(2^{height} * 2^{width} * (4+4)\right)$ bytes. Además, se utiliza una matriz de valores $bool$ de la misma dimensión que la cola de prioridad que posibilita el cálculo de las variables en el $Algoritmo$ 3.2.

Por el contrario, el cálculo exacto de la métrica está implementada en base a la estructura STL $unordered\_map <>$, la cual permite el conteo real de las frecuencias de los k-mer del dataset, con espacio lineal.

La Figura 4.16 muestra el espacio total utilizado en las ejecuciones de estimación de la Distancia Jensen-Shannon, y en la Figura 4.17 se muestra el espacio utilizado en el cálculo exacto de la métrica comparado al espacio de estimación promedio utilizado.

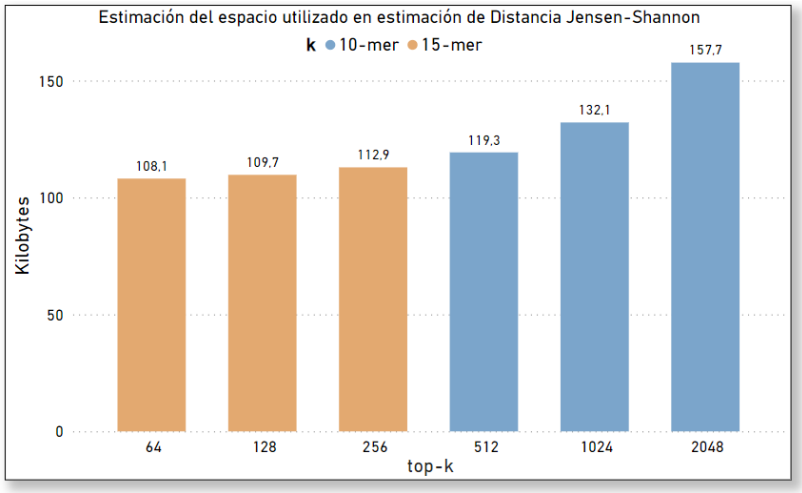


Figura 4.16: Estimación del espacio utilizado en la estimación de Distancia Jensen-Shannon, en casos 10 y 15-mer, en base a los valores de los parámetros mostrados en la Tabla 4.10.

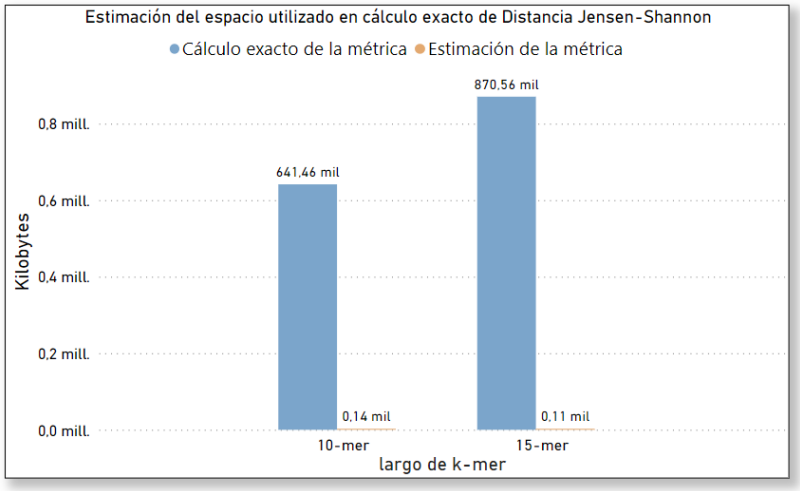


Figura 4.17: Estimación del espacio en el cálculo exacto de la Distancia Jensen-Shannon, en casos 10 y 15-mer. Además, se muestra el promedio del espacio utilizado por el método de estimación de la métrica en cada caso de k-mer, en base a los valores de parámetros de la Tabla 4.10.

De acuerdo con los gráficos, es claro que la diferencia entre el espacio utilizado en la estimación y el utilizado en el cálculo exacto. Esto se debe al uso de los sketches de estimación de frecuencias y cardinalidad, los cuales trabajan con una complejidad de espacio sublineal. El cálculo exacto, en cambio, debe realizar el conteo de las frecuencias de la totalidad de los k-mer, según su valor de cadena de texto, lo cual dispara el uso de la memoria.

A continuación, en la Figura 4.18 y Figura 4.19, se muestran el tiempo promedio de ejecución de estimación de la Distancia Jensen-Shannon, y el tiempo promedio de ejecución del cálculo exacto de la métrica, respectivamente.

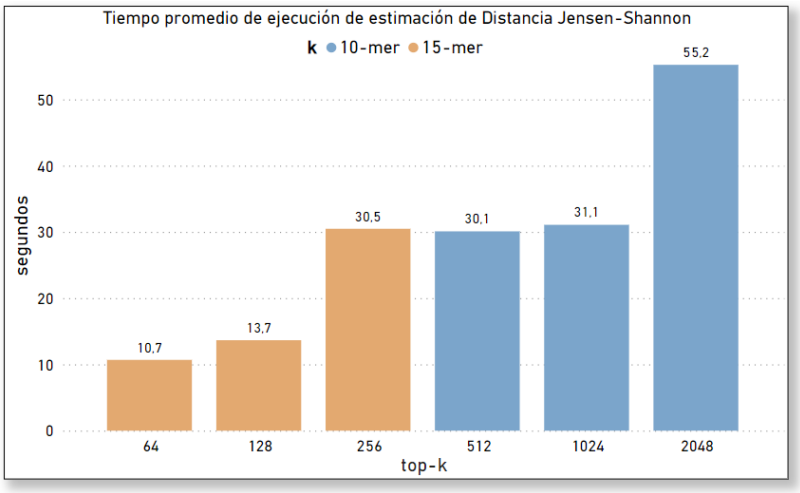


Figura 4.18: Tiempo promedio de ejecución en estimación de Distancia Jensen-Shannon, en casos 10 y 15-mer, en base a los valores de los parámetros mostrados en la Tabla 4.10.

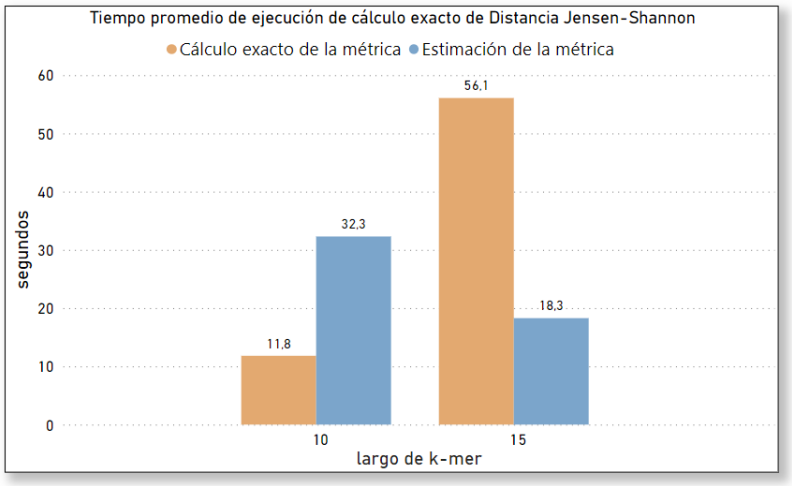


Figura 4.19: Tiempo promedio de ejecución en cálculo exacto de la Distancia Jensen-Shannon en casos 10 y 15-mer. Además, se muestra el promedio del tiempo de ejecución de estimación de la métrica, en base a los valores de los parámetros de la Tabla 4.10.

De acuerdo con la Figura 4.18, los tiempos de ejecución de estimación de la métrica tienden a aumentar a medida que se consideran más elementos top-K, y a la vez el largo del k-mer también condiciona el tiempo de ejecución de la estimación. Además, de acuerdo con la Figura 4.19, el tiempo promedio de ejecución es desfavorable en el caso 10-mer. En efecto, el tiempo de ejecución de la estimación está condicionado también por el largo k de la secuencia.

Conclusiones

Los objetivos propuestos en el desarrollo de este trabajo se cumplen satisfactoriamente. En primer lugar, se identifican los algoritmos basados en sketches del trabajo [8], y se adaptan junto con el método de estimación de entropía propuesto, esto en el contexto del análisis y comparación de conjuntos de secuencias genómicas. Con respecto a esto, se puede concluir que los resultados de estimación de entropía H son mejores en el caso 15-mer, esto debido a las características de la distribución de las frecuencias de los k-mer en los histogramas de cada caso. En efecto, el método de estimación de entropía H, es más exacto en los conjuntos de secuencias de genomas cuya entropía queda mejor representada por la cantidad de top-K considerados.

Luego, se implementó la estimación de la Distancia Jensen-Shannon, mediante el método de estimación de entropía H, y se realizaron pruebas para observar la precisión de la métrica. Con respecto a esto, en el caso 15-mer, los resultados se condicen con la tendencia de los resultados de la estimación de entropía, es decir, a menor cantidad de secuencias top-K consideradas, la métrica de distancia estimada es más exacta (Figura 4.13). Sin embargo, ocurre la situación contraria en el caso 10-mer, es decir, la tendencia de los resultados de estimación de entropía no se ve reflejada en los resultados de la estimación de la métrica. En el caso 10-mer, la estimación de H es más exacta a medida que se aumenta la cantidad de secuencias top-K consideradas, pero el $RMSE$ de estimación de la métrica de distancia en el caso 10-mer, aumenta a mayor cantidad de top-K considerados. Esto se puede explicar por la sobreestimación de las secuencias más frecuentes, lo cual condiciona también la exactitud de la estimación de entropía de la combinación de ambos conjuntos $H(M_{AB})$, y lo que en definitiva resulta en estimaciones más inexactas de la métrica.

Por otro lado, la comparación entre los tiempos de ejecución de estimación de la métrica y el cálculo exacto entrega resultados positivos. Sin embargo, el tiempo de ejecución de la estimación también está condicionado por el caso de longitud de k-mer. En efecto, los tiempos de ejecución de estimación de la métrica son desfavorables a menor longitud de k-mer. También, se confirma el espacio sublineal de los sketches en las estimaciones realizadas en ambos casos de longitud de k-mer.

Glosario

- Alineación: En la bioinformática es un método de organización de secuencias genómicas, el cual facilita la representación y comparación de las secuencias de ADN, ARN o estructuras proteicas.

- Análisis genómico: Ciencia que busca estudiar los genomas de los organismos, posibilitando la comparación de rasgos de diversa índole.

- Bit más significativo: En una representación de bits, es aquel que ocupa la posición más a la derecha, es decir, tiene el mayor peso. Se le conoce con las siglas MSB (Most Significant Bit). Para el caso del Least Significant Bit el bit se ubica en la posición más a la izquierda.

- Bucket: Espacio de memoria del sketch que permite almacenar algún tipo de dato en particular. Se refiere, por ejemplo, a un espacio de un arreglo, un vector, o una matriz.

- Cardinalidad: En el contexto de este trabajo la cardinalidad se define como la cantidad de elementos distintos en un conjunto. Por ejemplo, para un conjunto de secuencias genómicas, la cardinalidad del conjunto corresponde a la cantidad de secuencias distintas presentes en el conjunto.

- Conjunto de secuencias genómicas: Es un conjunto de secuencias k-mer generadas por los nucleótidos A, C, G, T, los cuales componen a los cromosomas de un individuo o especie. Pueden corresponder a la secuenciación del genoma completo, o a un subconjunto del genoma (Se recomienda ver Anexo 2).

- Flujo de elementos: Son aquellos datos, o elementos en el contexto del trabajo base referenciado [9], los cuales son procesados en streaming de manera continua y a una alta velocidad.

- Genoma de ADN: Secuenciación total de ADN de un organismo en particular (Anexo 2).

- Heavy Hitters: Son los elementos que aparecen con una frecuencia superior a un umbral predefinido en un flujo de datos. El problema de los Heavy Hitters es similar al problema de los top-K, en el que el objetivo es buscar los elementos más frecuentes de un conjunto de elementos. Sin embargo, sus aplicaciones son distintas.

- k-mer (secuencia k-mer): Subcadenas de largo k contenidas en una secuenciación genómica. En el contexto de esta memoria se utilizan las secuencias 10-mer y 15-mer.

- Leading zeros: En el contexto de este trabajo, corresponden a la cantidad de dígitos 0 ubicados antes del primer digito 1, contando desde el bit menos significativo (LSB).

- Libres de alineación: Los métodos libres de alineación son todos aquellos métodos que no producen ningún tipo de alineación, al intentar cuantificar la similitud entre secuencias genómicas.

- Parámetro de precisión del sketch: Es un parámetro que define la dimensión de un sketch en particular. Se dice que es de precisión, porque la elección del parámetro se relaciona directamente al rendimiento del sketch en cuanto a la precisión del cálculo de la estimación de una función de interés.

- Sketch: Un sketch es una estructura de datos reducida en espacio, usada para mantener datos que se utilizan para evaluar y/o calcular una estimación de una función de interés. Los sketches permiten generar un conjunto de datos más compacto que el original, aceptan consultas con baja complejidad lineal y con espacio sub-lineal, lo que permite optimizar el uso de los recursos, pero están sujetas a un error de estimación. Son utilizados para diversas aplicaciones donde se dispone de pocos recursos de memoria o se desea implementar la resolución de consultas en línea, y son aplicados también, por ejemplo, en el contexto de procesamiento de señales, en problemas de reducción de dimensionalidad, entre otros.

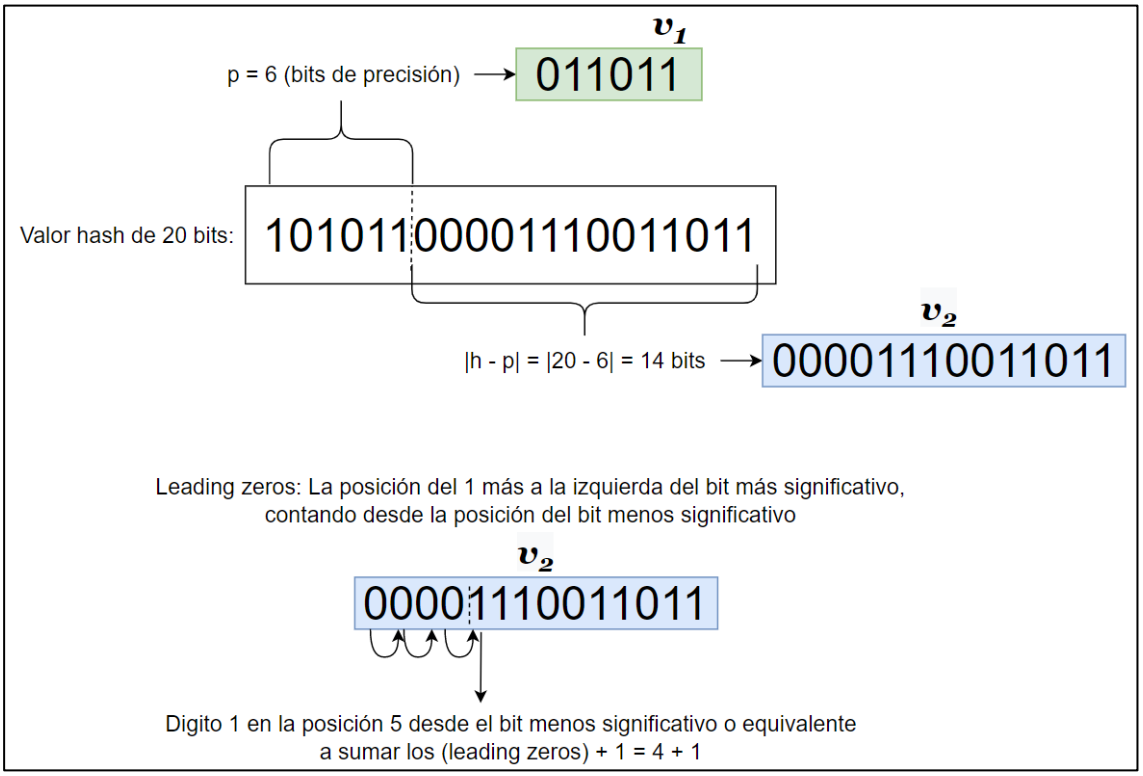
- top-K: En el contexto de este trabajo es el conjunto de las secuencias genómicas k-mer con mayor frecuencia dentro del conjunto de todas las posibles secuencias k-mer de un genoma. Por ejemplo, las 128 secuencias 10-mer más frecuentes dentro de una secuenciación de genoma se denominan top-K, con K=128.

Anexos

Anexo 1:

Los valores v_1 y v_2 en el algoritmo HyperLogLog se obtienen mediante la observación de patrones de bits. Esto es en tiene relación con la posición del dígito 1 más a la izquierda de los b bits mas significativos del valor hash $h(e)$ del elemento e procesado.

En el ejemplo gráfico de la Figura 2.1 se utiliza un valor hash de 20 bits, y un parámetro de precisión del sketch p=6. Es decir, para v_1, se utilizan los 6 bits menos significativos para establecer la posición en el vector A y, luego, se utilizan los $|h(e) - p|\ bits$ más significativos para el conteo de los leading zeros. Los leading zeros corresponden a la cantidad de dígitos 0 ubicados antes del primer digito 1, contando desde el bit menos significativo (LSB). Esto se detalla a continuación de manera gráfica.

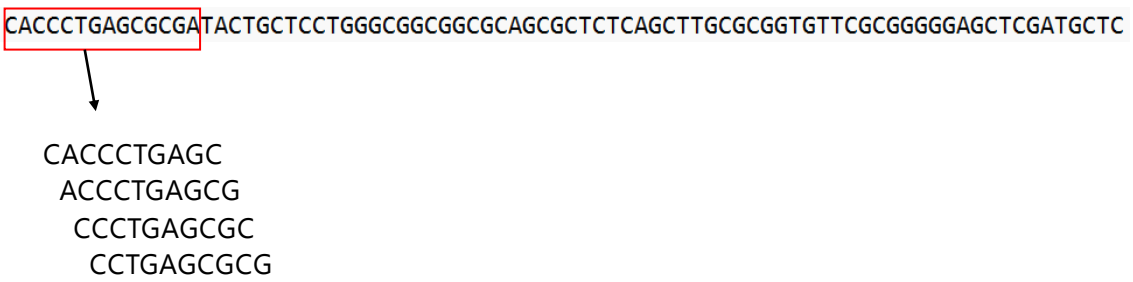


Anexo 2:

En la imagen se muestra el ejemplo de una porción pequeña de un genoma completo de ADN de la especie Minicystis rosea strain (200.510 líneas de secuencias del genoma).

```
>NZ_CP016211.1 Minicystis rosea strain DSM 24000 chromosome, complete genome
GATGCGACCTCGCGGCCGAGCCTGCGATCGCGCGCCAAGAACACGCGGCCCATCGCGCCCTCGCCGAGGAAGCGCACAGG
CTCGTACTTGTCCCAATGTGCGACGGGGAACGCTGGGCCTGCGGCCTCGCGCAGGCCGGAGGGCGGCGCCGAAGACCGCG
AGAGCGTGGCGTCGAGATCGGCCCTCTGCGGGAGCGTGGCGTCAGCGCCTGCCCGAGGATCCGCCGCGTCGAGGTGGACC
TTCACGCCCATCGCACGGGCGCGAGGATAACGAAGAGGATGGTTGAGGAGGAAGCCTGCCCCGTGCGCCCTGCCGGCTTT
CTTCGTCTGGGCGAGGAAGTCCCGGCTGCAGTCGTGGCTGCGGGCGCGACCGCGGGCGGGGCGAGGCGCTTCACGGAGCG
CGATGTGGTGAGGTCCGCGCACGAGTCACGCCCCGTCACGCATGCTGGGCGACTGCTGCCGTCGCGCTCGTGCGCCTCCC
GCGCGCGAGCGCGTAGCCAGCGCAAACGGGTTGGCAAAGTCGAGCCCAAACCAGGCCGGGATCAGGCGGCCAGTCTCCCA
GTTGATCATGGCCGGGGACGGCGTCGACGGCTCCGCGGCCGTCGTGATCAGGTCTTCGAACGGCGGCATGTATCCCATCG
ATCAACCGGATCGAGGACTGACACTTCGCATCCATGGAGTTCGATGCCGAGCGCGCCAGTGAAGATCTGGCGTGCTGACT
GCCGCAGCCAATGAGCCGCGTGCAGGACGTCACGACTGGCCCGCGAGGGCGCCCTCACCTGCACGCTGACGGGCTGGAAC
GTCTACGCCGACATGCGCGGCTCGACGCTCGCCTCGCGGTTCTTCGAGGAGCGCTCGCGAAGAGGGGCGCCGGCGAGCGC
ATAGGCACCCTCGCCGCGCTCGCGAGCTCCGCAGCCGTGTGGATGGGCGGCGGGCGGCGGCGCGCCATCGGCGCAGGCCG
CGAGCGCGAGGGCCGCGATCAGAGGGCGAGCATGGATCGCGCGGGGAAGGTGACGATCATGGCGACACGACGCATGCGCG
GAGGCCGGCGGCGGGGCGCCGGCAATCCTCCTCGGTCTTGCGACGGGCGGGGGGCGGGCACCATGGGATCCGCATGCCCG
CGCGTGCGGCGTGCGCTGGCGGGGAGGAACCGACATGGGCGTGCGACGGGATCATCGGGAGATCGCGTGGACCGTCGGCG
CGCTGCTCGGCGCGGTGGCCGCGGCGTACGCGGGGCCGCGCCGAGGGGGGCGCCGGCCTGATGCGCGCGTTCGCCGCGAT
GTACCGACTTCGACCGCCAACCACCGTCACGACCTGAAGGAGTCCATGACATGAGCCAGATCCTCGCCAACATTGCCGAC
GTTCGCATCCCCGACACCCCGTTGGTCCGCGAGGCCACCGAGCTGGTCCGGGCGGCGACCAGCGATCTCTTGTTCGACCA
CTCCAGCCGGGTGTTCCTCTGGGGCACGTTGAAGGGCCGCGCCCGTGGGCTGACCGCCGACCCCGAGTTGCTCTACGTGG
GCGCCATGTTCCACGACCTCGGCCTGACCGAGACGTACGGGCGCAAGGACCAGAGATTCGAGATCGACGGCGCGGACGCC
GCCCGCGACTTCCTCCTCGCGCACGGCTACAGCGCACAGGACGCACGGACGGTGTGGCTCTCGATCGCCCTGCACACCAC
CCCCGAGGTCCCCTATCACCTCGAGCCGGAGATCGCGCTGGTCACCGCCGGAGTGGAGACCGACGTCCTCGGCTTCTCAC
TCGAGGAGCTGACCGAGGAGCAGATCGCGCAGGTGGTCGCCGCGCATCCCCGCCCGGACTTCAAGCATCGCATCCTCGCC
GCGTTCTATGCCGGGATGAAGGACCGCCCGGACACGACGTTCGGAACCATGAATGATGACGTCCTCGCCCACTTCGACCC
GACCTTCGAACGCAAGGACTTCGTCGACATCATTCTCAGCTCCGCCTGGCCCGAATGATCGGCAGTACATGCGCAGGCGA
AGCTCCGCGCCTTCATCGACACGGCGCTCGCCTCCTGACTTCGCAGATGCGTTTGGACTCACCATGGGGCAGGCAGTGTA
TGCTGGTGCCGCTGAATCGATGGCTCACGAAGGAGATCGAACACTTGTCCCAGTGTTCGGTCTGGCGCCTCCCTCGGGAT
```

Luego, se muestra para el ejemplo 10-mer (k=10), algunas de las subsecuencias obtenidas de largo 10, las cuales se definen de la siguiente manera. Se consideran 10 nucleótidos contando desde el primer carácter, luego 10 nucleótidos contando desde el segundo carácter y así sucesivamente para el conjunto completo de líneas de genoma, es decir, el genoma completo. Por lo cual, se genera un conjunto de subsecuencias de largo 10, las cuales caracterizan de manera única el genoma completo de ADN. A este conjunto de subsecuencias de largo 10, lo denomino conjunto de secuencias genómicas, y son los datos utilizados en el contexto de esta memoria. También se le puede nombrar distribución de frecuencias de las secuencias 10-mer.



Bibliografía

[1] Altschul SF, Pop M. Sequence Alignment. In: Rosen KH, Shier DR, Goddard W, editors. Handbook of Discrete and Combinatorial Mathematics. 2nd edition. Boca Raton (FL): CRC Press/Taylor & Francis; 2017 Nov. Chapter 20.1. Available from: https://www.ncbi.nlm.nih.gov/books/NBK464187/

[2] McGinnis S, Madden TL. BLAST: at the core of a powerful and diverse set of sequence analysis tools. Nucleic Acids Res. 2004 Jul 1;32(Web Server issue):W20-5. doi: 10.1093/nar/gkh435. PMID: 15215342; PMCID: PMC441573.

[3] Pérez-Wohlfeil, E., Diaz-del-Pino, S. & Trelles, O. Ultra-fast genome comparison for large-scale genomic experiments. Sci Rep 9, 10274 (2019). https://doi.org/10.1038/s41598-019-46773-w

[4] Alignment-free sequence comparison benefits, applications, and tolos; https://genomebiology.biomedcentral.com/track/pdf/10.1186/s13059-017-1319-7.pdf

[5] Divergence measures based on the Shannon entropy - Information Theory, IEEE Transactions on (ufl.edu) https://www.cise.ufl.edu/~anand/sp06/jensen-shannon.pdf

[6] Mehul Jani and Rajeev K. Azad. 2013. Information entropy based methods for genome comparison. ACM SIGBioinformatics Rec. 3, 2, Article 2 (May 2013), 4 pages. https://doi.org/10.1145/2500124.2500126

[7] Guo Xuan. Frontiers in Genetics. 2020. Volume 11. 10.3389/fgene.2020.507038 JS-MA: A Jensen-Shannon Divergence Based Method for Mapping Genome-Wide Associations on Multiple Diseases https://www.frontiersin.org/articles/10.3389/fgene.2020.507038

[8] Feature frequency profile-based phylogenies are inaccurate Yuanning Li, Kyle T. David, Xing-Xing Shen, Jacob L. Steenwyk, Kenneth M. Halanych, and Antonis Rokas, November 24, 2020 https://doi.org/10.1073/pnas.2013143117

[9] A High-Throughput Hardware Accelerator for Network Entropy Estimation Using Sketches. J. Soto et al. IEEE Access, 2021

[10] Sketching and Sublinear Data Structures in Genomics https://www.annualreviews.org/doi/abs/10.1146/annurev-biodatasci-072018-021156

[11] RefSeq: NCBI Reference Sequence Database (nih.gov) https://www.ncbi.nlm.nih.gov/refseq/

[12] Baker DN, Langmead B. Dashing: fast and accurate genomic distances with HyperLogLog. bioRxiv. 2019:501726.

[13] Cormode, Graham (2009). "Count-min sketch". *Encyclopedia of Database Systems*. Springer. pp. 511–516.

[14] S10.pdf (rice.edu)
https://www.cs.rice.edu/~as143/COMP480_580_Spring19/scribe/S10.pdf

[15] Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016;17:132.

[16] Kullback, S.; Leibler, R.A. (1951). «On Information and Sufficiency». Annals of Mathematical Statistics 22 (1): 79-86. MR 39968. doi:10.1214/aoms/1177729694. [k-mer] van Dam RM, Quake SR. Gene expression analysis with universal n-mer arrays. Genome Res. 2002 Jan;12(1):145-52. doi: 10.1101/gr.198901. PMID: 11779839; PMCID: PMC155258.

[17] Sims GE, Jun SR, Wu GA, Kim SH. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. Proc Natl Acad Sci U S A. 2009 Feb 24;106(8):2677-82. doi: 10.1073/pnas.0813249106. Epub 2009 Feb 2. PMID: 19188606; PMCID: PMC2634796.

[18] Xu, Yi & Wasnik, Samiksha & Baylink, David & Berumen, Edmundo & Tang, Xiaolei. (2017). Overlapping Peptide Library to Map Qa-1 Epitopes in a Protein. Journal of Visualized Experiments. 2017. 10.3791/56401.

[19] Zhang Q, Jun SR, Leuze M, Ussery D, Nookaew I. Filogenómica viral utilizando un método sin alineación: un enfoque de tres pasos para determinar la longitud óptima de k-mer. Sci Rep. 2017 Enero 19;7:40712. doi: 10.1038/srep40712. PMID: 28102365; PMCID: PMC5244389.

[20] https://cplusplus.com/reference/functional/hash/

[21] https://sites.google.com/site/murmurhash/