Universidad de Concepción

UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

MEMORIA DE TÍTULO:

CRITERIO DE SELECCIÓN PARA REDUCIR EL NÚMERO DE COMPARACIONES DE SIMILITUD GENÓMICA UTILIZANDO SKETCHES

Autor: Álvaro Guzmán Chacón

Profesores Guía:

Julio Aracena Lucero
Cecilia Hernández Rivas

Agradecimientos

Se agradece a proyecto FONDECYT regular 1220960 por el apoyo financiero para la ejecución de la memoria.

Agradezco de manera especial y profunda a mi mamá, María Angélica, y a mi hermana, Judith Carolina. Han sido una fuente inagotable de amor, comprensión y apoyo incondicional a lo largo de mi carrera académica y de mi vida. Sus palabras de aliento y sacrificios han sido lo que me ha permitido llegar a donde estoy. Gracias por siempre estar ahí; no saben cuánto significa para mí saber que siempre puedo contar con ustedes. También quiero expresar mi agradecimiento a mi pequeño sobrino, Mateo Alonso, por ser una fuente de felicidad y alegría para mi familia.

A mis profesores guías, el Dr. Julio Aracena y la Dra. Cecilia Hernández, quiero expresar mi profunda gratitud. Gracias por su dedicación y orientación, que han sido cruciales no solo en el desarrollo de este trabajo, sino también en mi crecimiento como profesional. Aprecio enormemente su disponibilidad y apoyo constante, aspectos que siempre recordaré y valoraré.

Mis amigos y compañeros de estudio también merecen un agradecimiento profundo. Esteban, Francisco T., Claudio, Isidora, Iván y Francisco A., agradezco haber compartido con ustedes esta etapa de mi vida académica. También quiero agradecer a mis amigos de toda la vida de mi ciudad natal, Linares: Nicolás, Bastián, Rubén, Raúl, Gustavo y Simón. Agradezco a 'Los Viajeros' por su amistad y apoyo inquebrantable. Su amistad ha sido fundamental en mi desarrollo personal y académico.

También al profesor Miguel Figueroa, gracias por la oportunidad de pertenecer al laboratorio VLSI y por la ayuda que me ha dado durante este trabajo.

La comunidad académica en su conjunto también merece un reconocimiento especial. Agradezco la disposición constante de los jefes de carrera que me acompañaron a lo largo de mis estudios, la profesora Mónica Selva y el profesor Dominique Sphener. También quiero expresar mi agradecimiento a los funcionarios de la facultad, como la Sra. Cecilia y la Sra. Paola, quienes siempre mostraron una voluntad inmensa para ayudar.

Por último, a mi pareja, Beatriz Vera, quien ha sido mi compañera durante casi toda mi etapa universitaria. Nuestra historia es larga y el apoyo que he recibido de ti es invaluable. Siempre estuviste a mi lado, permitiéndome superar los desafíos que la vida nos presentaba. Te amo profundamente.

Esta memoria no habría sido posible sin el respaldo y la colaboración de todas estas personas. Mi gratitud es profunda y será para siempre.

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

1.1. Motivación

Comparar la similitud entre conjuntos de datos de ADN es una herramienta muy importante en el campo de la biología, con aplicaciones en el estudio de la relación entre bacterias [8] y, en general, en estudios de organismos biológicos [7]. Una medida muy utilizada para establecer similitud entre secuencias de ADN es la identidad nucleótida media (ANI por sus siglas en inglés) [10]. Por lo general, un ANI de un 95 % entre secuencias indica una relación fuerte.

Computar el ANI requiere de alineamiento de secuencias, lo cual es muy costoso computacionalmente. Además, las tecnologías modernas de secuenciación masiva han permitido la generación de colecciones grandes de bases de datos genómicas que requieren de algoritmos escalables que permitan su procesamiento [17, 18, 19]. Por este motivo surge la necesidad de desarrollar nuevas técnicas capaces de realizar estimaciones de medidas de similitud y que puedan manejar la gran escala de cantidad datos actual.

En este contexto aparecen las estructuras probabilistas llamadas **sketches**, estas corresponden a estructuras de datos que ocupan espacio limitado (típicamente sublineal) que permiten estimar alguna función de interés de algún conjunto de datos.

Por ejemplo, MinHash [3] e HyperLogLog [9] son sketches que estiman la similitud entre conjuntos y la cardinalidad de conjuntos, respectivamente. Gracias a estas estructuras es que aparecen herramientas como MASH [15] y Dashing [1]. A pesar de utilizar sketches distintos, ambas herramientas comparten el objetivo de estimar similitud genómica.

Sin embargo, cuando se tiene una gran base de datos, una herramienta como Dashing calcularía la similitud genómica entre todos los pares posibles de secuencias y, en general, uno necesita sólo pares de secuencias que cumplan con una similitud mínima, como el ANI del 95 % mencionado al inicio. Esto lleva a la pregunta, ¿es posible reducir el número de comparaciones utilizando algún criterio de selección en base a las restricciones?

El índice de Jaccard [16] es un índice de similitud entre dos conjuntos que toma valores entre 0 y 1, donde 0 indica una baja similitud (conjuntos disjuntos) y 1, una gran similitud (conjuntos iguales). Este coeficiente se denota por $J(A, B)$, donde A y B son los conjuntos a comparar. Dado que una secuencia de ADN se puede representar como un conjunto de sub-secuencias de largo k, llamado conjunto de k-mers [13], se puede estimar la similitud entre dos secuencias a través del correspondiente índice de Jaccard entre los conjuntos de k-mers. Este índice es muy relevante en el estudio de similitud genómica, pues presenta una fuerte correlación con el ANI y la distancia MASH [10, 15]. Así, una forma de pedir cierta similitud entre dos conjuntos A, B, sería pedir que $h \leq J(A, B)$, para algún h entre 0 y 1. Luego, si se tienen N secuencias, comparar a todos los pares resulta en $N(N-1)/2$ comparaciones, es decir, es $\mathcal{O}(N^2)$ con respecto a la cantidad de secuencias. Dado que las bases de datos crecen rápidamente, es necesario encontrar métodos que permitan agilizar al proceso de comparación.

Buscar sólo los pares que cumplan con tener un coeficiente de Jaccard mayor a h permite encontrar un criterio de selección determinista dado por las cardinalidades de los conjuntos en cuestión. Este criterio [17] logra reducir el número de comparaciones entre conjuntos de k-mers que se realizan, al costo mínimo de comparar sólo las cardinalidades individuales de ambos conjuntos. Acelerando así el proceso

de selección. El objetivo del presente estudio es analizar este criterio de selección directa y definir un nuevo criterio que permita reducir el número de comparaciones a través de una comparación barata entre los conjuntos de k-mers, y así reducir el tiempo de cómputo.

1.2. Contenido del documento

El documento se desarrollará de la siguiente manera. En el capítulo 2 se encuentran las definiciones y conceptos que son necesarios para el trabajo realizado, además de las notaciones y convenciones a utilizar. El capitulo 3 busca explicar métodos y estrategias ya existentes para computar similitud entre secuencias. En el capítulo 4 es donde se desarrolla la idea de un nuevo criterio de selección, abarcando el respaldo teórico, garantías y definición de este nuevo criterio. Finalmente, en el capítulo 5 se discuten los resultados obtenidos y la utilidad práctica del criterio propuesto.

1.3. Objetivos

El objetivo de esta memoria es proponer y analizar criterios de selección para reducir el número de comparaciones de similitud genómica cuando se necesita seleccionar pares de secuencias cuyo coeficiente de Jaccard sea mayor a un umbral $h \in (0, 1)$. Los objetivos específicos son:

O1: Proponer un criterio de selección basado en un análisis de error para la estimación del coeficiente de Jaccard dada por sketches de Hyperloglog.

O2: Implementar y evaluar el criterio propuesto con conjuntos pequeños de datos de la base de datos RefSeq[1].

O3: Comparar los resultados del criterio propuesto experimentalmente con el criterio ya existente.

[1] https://www.ncbi.nlm.nih.gov/refseq/

Capítulo 2

Definiciones y conceptos previos

2.1. Notación y definiciones

1. Un multiconjunto, también conocido como bolsa o conjunto con repetición, es una generalización de la idea de conjunto que permite la existencia de elementos repetidos. A diferencia de un conjunto convencional, donde cada elemento puede aparecer solo una vez, en un multiconjunto un elemento puede repetirse varias veces.

   Formalmente, un multiconjunto se puede definir como un par $M = (A, m)$, donde A es el conjunto subyacente del multiconjunto, formado por los elementos distintos de M. Y $m : A \to \mathbb{N}$ es una función que mapea cada elemento $a \in A$ a su multiplicidad (número de ocurrencias de a en M) $m(a)$. La cardinalidad de M (i.e. la cantidad de elementos distintos de M) se denotará por $|M|$. Notar que $|M|$ coincide con la cantidad de elementos distintos de A ($|M| = |A|$).

   Más explícitamente, el multiconjunto $M = (A, m)$ se representará como el siguiente conjunto:

   $$M = \{(a, m(a) \ : \quad a \in A)\}.$$

   Por ejemplo, $M = \{1, 1, 2, 5, 3, 2, 1, 5, 4\}$ es un multiconjunto, el cual repre-

sentaremos como $M = \{(1,3),(2,2),(3,1),(4,1),(5,2)\}$, ya que el 1 aparece 3 veces, el 2, 2 veces, y así hasta el 5 que aparece 2 veces. Además, su cardinalidad es $|M| = 5$.

2. Σ denotará al alfabeto de los nucleótidos (unidades fundamentales del material genético), es decir, $\Sigma = \{A, C, G, T\}$. Denotaremos por Σ^k al conjunto de todas las palabras de largo k sobre Σ, y por Σ^*, al conjunto de todas las palabras de largo finito sobre Σ (sin importar su largo). Es decir, Σ^k corresponde a todas las cadenas de k nucleótidos y Σ^*, al conjunto de secuencias genómicas. Además, diremos que $w \in \Sigma^*$ es sub-palabra de $L \in \Sigma^*$, si los símbolos de w aparecen en L de forma contigua y en el mismo orden (notar que el largo de w no puede ser superior al de L). Por ejemplo, $L = ACTGCCGT$ es una secuencia genómica de largo 8 y $w = CTG \in \Sigma^3$ es una sub-palabra de L ya que aparece entre el segundo y cuarto símbolo.

3. Dados una secuencia genómica $L \in \Sigma^*$, y un número $k \in \mathbb{N}$, un k-mer de L corresponde a una sub-palabra $a \in \Sigma^k$ de L. El conjunto de k-mers de L se denota por $D_k(L)$. Por otro lado, el multiconjunto de k-mers de L, que denotaremos por $spec_k(L)$, corresponde al multiconjunto $(D_k(L), m)$, siendo $m : D_k(L) \to \mathbb{N}$ la función que retorna la multiplicidad de cada k-mer en L. Es decir,

$$spec_k(L) := \{(a, m(a)) \ : \quad a \in D_k(L)\}.$$

La cardinalidad de este multiconjunto se denotará por $d_k(L)$. Por ejemplo, si $L = ATTTCGTC$, entonces

$$\begin{aligned} spec_2(L) &= \{AT, TT, TT, TC, CG, GT, TC\} \\ &= \{(AT,1),(TT,2),(TC,2),(CG,1),(GT,1)\}, \end{aligned}$$

y $d_2(L) = 5$, ya que sólo aparecen 5 k-mers distintos. En la Figura 2.1 se muestra un ejemplo ilustrativo de 4-mers para una secuencia de largo 11.

Notar que si L es una secuencia genómica de largo $N \in \mathbb{N}$, entonces, para todo $k \in \mathbb{N}$ se cumple que:

$$d_k(L) \leq \text{mín}\{4^k, N - k + 1\}.$$

Es evidente que $d_k(L) \leq 4^k$, pues $D_k(L) \subset \Sigma^k$ y $|\Sigma^k| = 4^k$. Además, como la secuencia L es de largo N, entonces contiene $N - k + 1$ sub-palabras de largo k (no necesariamente distintas), así que $spec_k(L) \leq N - k + 1$.

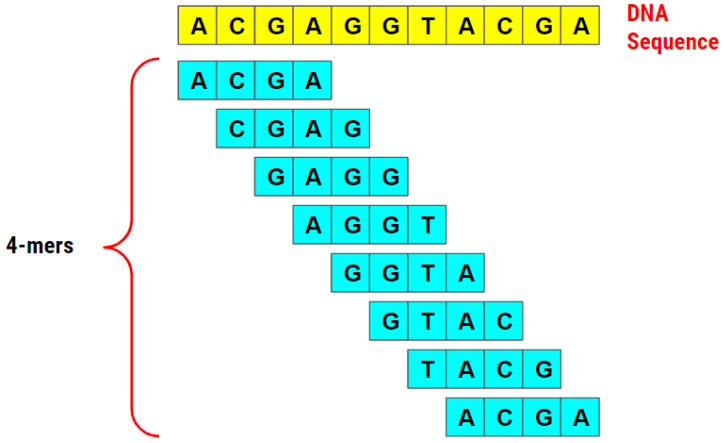

Figura 2.1: Ejemplo de 4-mers de una secuencia de largo 11.

4. Dados dos multiconjuntos A, B, el coeficiente de Jaccard entre A y B, denotado por $J(A, B)$, corresponde a la porción de elementos en común entre A y B, sin contar sus multiplicidades. Esto es,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A| + |B| - |A \cup B|}{|A \cup B|},$$

donde la segunda igualdad está dada por el principio de inclusión-exclusión ($|A \cup B| = |A| + |B| - |A \cap B|$).

Para aplicaciones basadas en distancias en lugar de similitud, el coeficiente de Jaccard puede ser transformado a la distancia de Jaccard $J_\delta = 1 - J$ [12].

En el caso particular de secuencias, dadas dos secuencias $L_1, L_2 \in \Sigma^*$, $J_L(L_1, L_2, k)$ denota el coeficiente de Jaccard entre $spec_k(L_1)$ y $spec_k(L_2)$. Es decir,

$$J_L(L_1, L_2, k) = J(spec_k(L_1), spec_k(L_2)) = \frac{|spec_k(L_1)| + |spec_k(L_2)| - |spec_k(L_1) \cup spec_k(L_2)|}{|spec_k(L_1) \cup spec_k(L_2)|}$$
$$= \frac{|D_k(L_1)| + |D_k(L_2)| - |D_k(L_1) \cup D_k(L_2)|}{|D_k(L_1) \cup D_k(L_2)|}$$

5. Dado $A = (a_1, a_2, ..., a_n) \in \mathbb{R}^n_+$, $H(A)$ denotará la media armónica entre los elementos de A. Es decir,

$$H(A) = \frac{n}{\sum_{i=1}^{n} a_i^{-1}}.$$

La media armónica cumple con la siguiente desigualdad:

$$\text{mín}\, A \leq H(A) \leq n\, \text{mín}\, A. \tag{2.1}$$

6. Dado $A = (a_1, a_2, ..., a_n) \in \mathbb{R}^n$, denotaremos $A[i] = a_i$. También denotaremos,

$$2^A := (2^{a_1}, 2^{a_2}, ..., 2^{a_n}) \in \mathbb{R}^n.$$

2.2. Teorema de Taylor

El teorema de Taylor es un teorema que permite aproximar una función en un punto donde sea diferenciable por medio de un polinomio.

Teorema 2.1 (Teorema de Taylor)**.** *Sea $n \in \mathbb{N}$ y sea $f : \mathbb{R} \to \mathbb{R}$ una función diferenciable n veces en el punto $a \in \mathbb{R}$. Entonces existe una función $h_n : \mathbb{R} \to \mathbb{R}$ tal que*

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + ... + \frac{f^{(n)}(a)}{n!}(x-a)^n + h_n(x)(x-a)^n,$$

con $\lim_{x\to a} h_n(x) = 0$.

El polinomio que aparece en el teorema se denomina polinomio de Taylor de orden n de la función f en el punto a, y se denota por $P_n(x)$. Es decir,

$$P_n(x) = \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i,$$

siendo $f^{(0)} = f$.

El término del resto, denotado por $R_n(x)$, corresponde al error de aproximación de $f(x)$ dado por $P_n(x)$, es decir,

$$R_n(x) = f(x) - P_n(x).$$

Suponiendo que f es continuamente diferenciable $n+1$ veces en un intervalo I que contiene a a, y suponiendo que existen constantes q y Q tales que

$$q \leq f^{(n+1)}(x) \leq Q$$

en el intervalo I. Entonces se cumple que:

$$q\frac{(x-a)^{n+1}}{(n+1)!} \leq R_n(x) \leq Q\frac{(x-a)^{n+1}}{(n+1)!}.$$

En particular, si

$$|f^{(n+1)}(x)| \leq M,$$

sobre un intervalo $I = (a-r, a+r)$, con $r > 0$, entonces

$$|R_n(x)| \leq M\frac{|x-a|^{n+1}}{(n+1)!} \leq M\frac{r^{n+1}}{(n+1)!}. \tag{2.2}$$

2.3. Sketches y streaming

Sea X un conjunto de datos y $f(X)$ alguna función de interés que se quiera estimar, como por ejemplo la media de X. Un sketch $S(X)$ de X es una estructura de datos, generalmente probabilística, que resume a X de forma que se pueda estimar $f(X)$ a partir de $S(X)$. Esta estructura de datos utiliza espacio acotado, típicamente sublineal.

Dado que el objetivo es estimar $f(X)$, estas estructuras de datos suelen justificarse a través de la teoría de la probabilidad. Uno de los primeros algoritmos de esta clase [5] utiliza variables geométricas para estimar la cantidad de elementos distintos en un conjunto X. Esta idea es justamente la que entrega la intuición para el sketch principal que se revisará en esta memoria: el sketch de Hyperloglog.

Teniendo X, siempre es posible obtener $f(X)$ de forma exacta si no se tienen restricciones del espacio ocupado. Sin embargo, los conjuntos de datos hoy en día pueden llegar a ser gigantescos, además de estar actualizándose constantemente. Por ejemplo, el tráfico de un sitio web como Google puede rondar los 80 billones mensuales en la actualidad. Realizar mediciones sobre conjuntos tan grandes puede requerir

mucho espacio. En el caso de calcular cardinalidad (cantidad de individuos distintos que visitan Google), se necesita almacenar cada elemento distinto. Tanto la cantidad de memoria necesaria, como el acceso a esta, provocan que este cálculo sea muy costoso computacionalmente.

El concepto de streaming viene de la idea de construir $S(X)$ realizando una pasa secuencial sobre los elementos de X, además, X suele ser un conjunto que se actualiza constantemente. Así, es deseable poder actualizar $S(X)$ a medida que nuevos elementos son agregados a X. En la literatura los sketches son frecuentemente asociados a una clase de algoritmos de streaming.

2.3.1. Funciones de hash

El hashing es una técnica clásica para insertar, borrar y buscar con tiempo promedio constante [4] . La idea es construir una función de hash $h : \mathcal{D} \to \{0, 1, ..., m-1\}$ que mapee los objetos de su universo original $\mathcal{D}$ a un valor en $\{0, 1, ..., m-1\}$. Así, los elementos $x \in \mathcal{D}$ se corresponden inequívocamente a un índice en $\{0, 1, ..., m-1\}$, a esta correspondencia se le conoce como tabla de hash. De esta forma, buscar un elemento en esta tabla es tan costoso como computar su función de hash, lo cual en la práctica es $\mathcal{O}(1)$.

Un problema que pueden presentar las funciones de hash son las colisiones, esto es, cuando dos valores distintos $x, y \in \mathcal{D}$ se mapean al mismo índice ($h(x) = h(y)$). Esto provoca que existan elecciones de elementos en $\mathcal{D}$ de forma que ocurran muchas colisiones, perdiéndose las garantías de tiempo $\mathcal{O}(1)$. Para lidiar con esto, en la práctica se utiliza el hashing universal.

Definición 2.1 (Hashing Universal [4])**.** *Sea $\mathcal{D}$ nuestro universo finito de elementos. Sea también $\mathcal{H}$ una familia de funciones hash que mapean los elementos de $\mathcal{D}$ a $\{0, 1, ..., m-1\}$, con $m \in \mathbb{N}$. Se dice que $\mathcal{H}$ es familia universal de funciones de hash si para todo par $x, y \in \mathcal{D}, x \neq y$, se tiene que*

$$Pr(h(x) = h(y)) \leq \frac{1}{m},$$

donde la probabilidad se toma con respecto a las posibles elecciones de funciones

$h \in \mathcal{H}$ (eligiendo al azar y uniformemente).

De esta forma, al tomar aleatoriamente una función de hash $h \in \mathcal{H}$, la probabilidad de colisión entre dos elementos distintos $x, y \in \mathcal{D}$ no es más que la probabilidad de colisión si $h(x)$ y $h(y)$ se hubieran escogido aleatoria e independientemente sobre los valores $\{0, 1, ..., m-1\}$. Lográndose así un control sobre las colisiones.

2.3.2. Sketch de Hyperloglog

Intuición

Consideremos el siguiente experimento: lanzar una moneda hasta obtener un sello y contar la cantidad de lanzamiento realizados. Notar que este experimento corresponde a una variable aleatoria discreta con distribución geométrica. Así, la probabilidad de que el primer sello se obtenga en la posición i es de 2^{-i}. Es decir, para observar este resultado, se espera que el experimento se haya realizado 2^i veces. De esta forma, observar la posición del primer sello que se obtiene permite estimar la cantidad de veces que se ha realizado el experimento. Además, al realizar varias veces este experimento, el resultado menos probable es el que controla la cantidad de veces que se ha realizado el experimento, ya que es el resultado más difícil de observar.

Por otro lado, sea X un conjunto de cardinalidad n, las funciones hash permiten mapear cada elemento de X a una sucesión binaria, que es equivalente al experimento de lanzar una moneda varias veces. Así, conectando el uso de las funciones hash con la idea del experimento de la moneda anteriormente presentado, es posible generar la intuición que hay detrás de la estimación de cardinalidad del algoritmo Hyperloglog.

Definición

Sea X un multiconjunto de cardinalidad $n \in \mathbb{N}$ con elementos de un universo $\mathcal{U}$. El sketch de Hyperloglog [6], [9] corresponde a un arreglo A de tamaño $m = 2^p$, con $p \in \mathbb{N}$, que se llena según se explica a continuación. Cada elemento x de X es procesado por una función de hash (de 32 bits para esta explicación) $h : \mathcal{U} \to \{0,1\}^{32}$. Después, $h(x)$ se divide en los primeros p bits, denotados por $v_1(x)$, y en

los últimos $b = 32 - p$ bits, denotados por $v_2(x)$. Así, $h(x)$ es la concatenación de $v_1(x)$ y $v_2(x)$.

De esta forma, los elementos de X se pueden dividir en m particiones. Sea $i \in \{0, 1, ..., m-1\}$, la partición i-ésima está dada por

$$P_i(X) = \{e \in X \ : \quad v_1(e) = i\}.$$

Si la función de hash se comporta suficientemente uniforme, entonces se espera que $|P_i(X)| = \frac{n}{m}$. Luego, se observa el valor de $v_2(e)$ a los elementos $e \in P_i(X)$ y, según la intuición dada, se toma la secuencia binaria resultante con mayor cantidad de ceros consecutivos al inicio (contando de izquierda a derecha) para estimar la cantidad de elementos en $P_i(X)$. Así, si A_i es la posición del primer 1 de esta secuencia, entonces la cantidad estimada de elementos distintos en $P_i(X)$ es 2^{A_i}. Formalmente,

$$A_i := \max_{e \in P_i(X)} \rho(v_2(e)),$$

donde $\rho : \{0,1\}^{32} \to \mathbb{N}_0$ es la función que indica la posición del primer 1, de izquierda a derecha, de una secuencia binaria. Así, por cada partición se obtiene el valor de A_i, y con él, una estimación de su cardinalidad. Finalmente, la estimación de $n = |X|$ está dada por

$$E = \alpha_m m H(2^A)$$

donde α_m es un factor de corrección y $A = (A_0, A_1, ..., A_{m-1})$ es el arreglo que representa al sketch de Hyperloglog. Es decir, $H(2^A)$ corresponde a la media armónica de los valores en el sketch. La media aritmética es sensible a los valores extremos, a diferencia de la media armónica que no lo es, es por ello que se utiliza esta última.

Para efectos prácticos, se utiliza una única función hash de 64 bits (o 32 bits) $h : \mathcal{U} \to \{0,1\}^{64}$ y esta se divide en 2, los primeros p bits y los últimos b bits. Los primeros p bits se utilizan para construir las particiones, generando $m = 2^p$ particiones, y los b bits siguientes se utilizan para llenar los registros del sketch A (Ver Figura 2.2).

Es claro que, $A_i = \mathcal{O}(\log_2 n)$, $\forall i \in \{1, 2, ..., m\}$. Además, almacenar los valores $A_i \in \mathbb{N}$ en memoria utiliza $\mathcal{O}(\log_2 A_i)$ bits, por lo que la cantidad de memoria necesaria para estimar n es $\mathcal{O}(\log_2 \log_2 n)$, de ahí que el algoritmo original se llame Loglog.

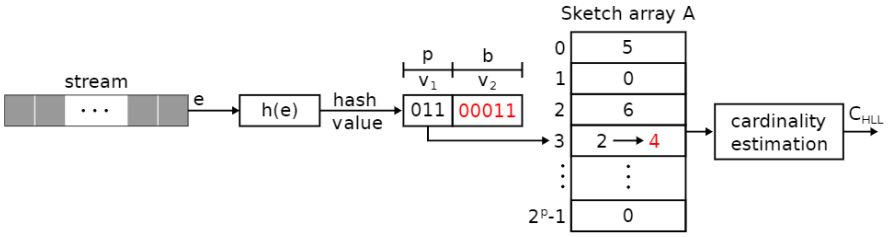

Figura 2.2: Esquema general de Hyperloglog. Se toman los elementos e pertenecientes a un multiconjunto o *stream* (flujo de datos) y se calcula su valor de hash $h(e)$. Este valor se separa en los primeros p bits que representan un valor v_1 y en los últimos b bits que representan un valor v_2. Luego, se actualiza la posición v_1 del sketch (o arreglo) A por el máximo entre el valor actual de $A[v_1]$ y el valor de $\rho(v_2)$. Una vez procesados los elementos del multiconjunto, se estima su cardinalidad utilizando los valores de A

El algoritmo anteriormente presentado, Hyperloglog, corresponde a una mejora del algoritmo Loglog, siendo el cambio principal el uso de la media armónica en lugar de la media aritmética.

El respaldo teórico de la estimación dada por este sketch está dado por el siguiente teorema:

Teorema 2.2. *Sea X un multiconjunto de cardinalidad n desconocida. Y sea E el resultado de aplicar el algoritmo de Hyperloglog a X utilizando $m \in \mathbb{N}$ registros según lo descrito anteriormente. Entonces*

1. *E es asintóticamente casi insesgado, esto es:*
$$\frac{1}{n}\mathbb{E}_n[E] \xrightarrow{n\to\infty} 1 + \delta_1(n) + o(1).$$
2. *El error estándar de E, dado por $\frac{1}{n}\sqrt{\mathbb{V}_n[E]}$, cumple lo siguiente:*
$$\frac{1}{n}\sqrt{\mathbb{V}_n[E]} \xrightarrow{n\to\infty} \frac{\beta_m}{\sqrt{m}} + \delta_2(n) + o(1).$$

Donde

- δ_1 *y* δ_2 *son funciones oscilantes de amplitud muy pequeña, aproximadas a 0 para efectos prácticos.*
- *La constante* β_m *es acotada, con* $\beta_{16} = 1.106$*,* $\beta_{32} = 1.070$*,* $\beta_{64} = 1.054$*,* $\beta_{128} = 1.06$*, y* $\beta_\infty = \sqrt{3\log 2 - 1} = 1.03896$*.*

Así, se espera que el estimador se acerque a n y, al aumentar el tamaño del sketch, se espera que el error estándar disminuya.

Denotaremos por $S_p(X)$ al sketch A de Hyperloglog de tamaño $m = 2^p$ asociado a X, el cuál es llenado según lo explicado anteriormente. Además, $|X|_p$ denotará la estimación de cardinalidad obtenida por este sketch.

Sean X, Y dos multiconjuntos, $x \in X$ e $i \in \{0, 1, ..., m-1\}$. Dada la forma en que se procesan los elementos de X para llenar $S_p(X)$, es posible establecer las siguientes propiedades del sketch de Hyperloglog:

$$i)\ |X \cup \{x\}|_p = |X|_p, \qquad \text{(Idempotente)}$$

$$ii)\ S_p(X \cup Y)[i] = \text{máx}\{S_p(X)[i], S_p(Y)[i]\}, \qquad \forall i \in \{1, 2, ..., m\}. \qquad \text{(Fusionable)}$$

Es decir, insertar más de una vez un elemento en el sketch tiene el mismo efecto en la estimación de cardinalidad que haberlo insertado una única vez. Claramente esta es una propiedad deseable para la estimación de cardinalidad. Por otro lado, la propiedad de fusión o merge, establece que a través de los sketches asociados a X e Y es posible construir el sketch asociado a $X \cup Y$, sin necesidad de procesar la unión como tal.

El error relativo estándar de un Hyperloglog de tamaño p (es decir, $m = 2^p$) es $\sigma_m = \beta_m/\sqrt{m}$, con β_m definido en el teorema 2.2. Según [6] se cumple lo siguiente: con una probabilidad de $\alpha \in [0, 1]$ se tiene que

$$|X|_p \in [|X| - Z_\alpha \sigma_m |X|, |X| + Z_\alpha \sigma_m |X|], \tag{2.3}$$

donde Z_α corresponde al valor tal que el $\alpha 100\,\%$ de los datos en una distribución normal estándar están en $[-Z_\alpha, Z_\alpha]$. Usualmente se usa $Z_\alpha = 1$, 2 y 3 para abarcar el 68.27 %, 95.45 % y 99.73 % de los datos, respectivamente.

Por comodidad, se usará la notación σ_p para referirse a σ_m. Es decir,

$$\sigma_p := \frac{\beta_{2^p}}{\sqrt{2^p}}.$$

Capítulo 3

Estado del arte

3.1. Comparación de genomas

3.1.1. MASH

MinHash es un sketch que recibe un multiconjunto de datos X y, por medio de funciones de hash, construye un arreglo (sketch) $S(X)$ para identificar a X. Luego, si se tienen los sketches de distintos multiconjuntos X_1, X_2, entonces comparar los sketches $S(X_1)$ y $S(X_2)$ permite computar una aproximación j de $J(X_1, X_2)$ (Ver Figura 3.1).

Mash [15] es una herramienta que utiliza como base el sketch de MinHash para procesar secuencias genómicas y calcular la distancia Mash. La llamada distancia Mash, denotada por D, entre dos secuencias $S_1, S_2 \in \Sigma^*$ está dada por

$$D(S_1, S_2) = -\frac{1}{k} \ln \frac{2j}{j+1}, \tag{3.1}$$

donde k es el parámetro para generar los k-mers y j es la estimación del coeficiente de Jaccard entre $spec_k(S_1)$ y $spec_k(S_2)$ entregada por los sketches de MinHash. D no es una distancia o métrica en el sentido matemático, pues no cumpla la desigualdad triangular.

Para la deducción de esta distancia se considera la probabilidad d de que ocurra una

mutación puntual (que una letra de la secuencia cambie) en un genoma w. Luego, bajo un modelo de Poisson, la probabilidad de que no ocurra ninguna mutación en un k-mer dado es de e^{-kd}, cuyo valor esperado corresponde a la fracción de k-mers no mutados w con respecto al número total de k-mers t en el genoma. Resolviendo $e^{-kd} = \frac{w}{t}$ resulta en $d = -\frac{1}{k} \ln \frac{w}{t}$. Para tener en cuenta el tamaño de ambos genomas a comparar, se considera t como el promedio de los tamaños de ambas secuencias, denotado por n. Finalmente, como la estimación del Jaccard j puede ser expresada en términos del tamaño promedio de los genomas $j = \frac{w}{2n-w}$, la fracción de k-mers compartidos puede ser expresada en términos de j, $\frac{w}{n} = \frac{2j}{j+1}$. Obteniéndose la distancia Mash (3.1).

Así, la distancia Mash D busca capturar la tasa de mutación para que de una secuencia se obtenga la otra. Por lo que un bajo valor de D indica que la tasa de mutación debe ser baja, es decir, las secuencias son muy similares. Por el contrario, si D es grande, entonces se infiere una alta tasa de mutación, indicando que las secuencias no son muy similares.

La relación entre el coeficiente de Jaccard J y la distancia Mash M es clara (D se define a partir de una estimación de J). Además, se muestra también en [15] que D se correlaciona también con el ANI. En específico, D se aproxima a $1-\text{ANI}$. Así, queda justificada la relevancia de estimar J o D para ahorrar recursos en la obtención del ANI.

3.1.2. Dashing

MinHash presenta problemas para estimaciones de cardinalidad al comparar secuencias con diferencias de tamaño muy grandes. Este escenario no es tan extraño. Por ejemplo al comparar el genoma de una bacteria, que es muy pequeño, a una colección de meta-genomas (muchos genomas relacionados).

Dashing [1] es una herramienta que utiliza el sketch de Hyperloglog para estimar la cantidad de k-mers distintos en una secuencia, y así estimar el coeficiente de Jaccard a partir de las cardinalidades correspondientes. Hyperloglog exhibe una excelente precisión y rapidez en una amplia variedad de escenarios, incluso cuando los

conjuntos de entrada son de tamaños muy diferentes o cuando los sketches correspondientes son pequeños.

Dadas dos secuencias L_1, L_2 y un valor $k \in \mathbb{N}$, y siguiendo la notación establecida en esta memoria, el coeficiente de Jaccard entre sus respectivos conjuntos de k-mers $D_k(L_1)$ y $D_k(L_2)$, se puede calcular como

$$J_L(L_1, L_2, k) = \frac{|D_k(L_1)| + |D_k(L_2)| - |D_k(L_1) \cup D_k(L_2)|}{|D_k(L_1) \cup D_k(L_2)|}. \tag{3.2}$$

Así, se puede construir los sketches de Hyperloglog de $S(spec_k(L_1))$ y $S(spec_k(L_1))$ asociados a L_1 y L_2, respectivamente. Luego, a partir de estos sketches se pueden estimar $|D_k(L_1)|, |D_k(L_2)|$ y $|D_k(L_1) \cup D_k(L_2)|$. Reemplazando estos valores en (3.2) se obtiene una estimación de $J(D_k(L_1), D_k(L_2))$. Dashing ocupa esta estrategia para computar estimaciones del coeficiente de Jaccard, además de ciertas correcciones al estimador de cardinalidad para mejorar el error en rangos de cardinalidad muy pequeños o muy grandes.

3.2. Cómo elegir k

Uno de los problemas claves del proceso de comparar dos secuencias genómicas es el de cómo elegir el valor de $k \in \mathbb{N}$ para generar los conjuntos de k-mers que se procesarán. Claramente una elección a ciegas puede llevar a resultados no significativos. Por ejemplo, si k es muy chico (por ejemplo $k = 1$), entonces $d_k(L)$ será muy cercano a 4^k para todas las secuencias L suficientemente largas. Por otro lado, si k es muy grande (cercano al largo de la misma secuencia por ejemplo), entonces $d_k(L)$ será muy cercano a 1.

3.2.1. Coeficiente de compresibilidad δ

Existen medidas que buscan comprimir secuencias capturando la repetividad que posean estas. Una de ellas es la medida δ [11].

Definición 3.1. *Sea una secuencia L de largo n, se define*

$$\delta(L) = \text{máx}\{d_k(L)/k \ : \ k \in \{1, 2, ..., n\}\}.$$

Se obtiene así una medida que es independiente a la elección de k. Al recorrer los distintos valores de k se pueden observar tres etapas en el comportamiento de $d_k(L)/k$. Para valores de k que son muy pequeños, de modo que casi todas las k-mers aparecen en L, $d_k(L)/k$ crece exponencialmente. Para valores de k cercanos a n, $d_k(L)/k$ decrece linealmente, eventualmente llegando a 1 cuando $k = n$. Para valores intermedios de k, ambos comportamientos anteriores están en tensión, incrementar k aumenta la cantidad de posibles k-mers, pero eventualmente en la secuencia dejan de aparecer muchos nuevos k-mers. El k que maximiza $d_k(L)/k$ identifica este punto en el que se deja de obtener mucha nueva información [2].

Si bien es costoso calcular $d_k(L)$ para varios valores de k, se pueden usar sketches para estimar su valor. Los autores de DanD [2] utilizan la herramienta de Dashing para construir los sketches y así calcular δ para distintas secuencias. DanD también ocupa esta medida de compresibilidad para proponer un nuevo coeficiente de similitud: el coeficiente de Jaccard independiente de k, KIJ por sus siglas en inglés.

Definición 3.2. *Sean L_1, L_2 dos secuencias. El coeficiente de Jaccard independiente de k entre L_1 y L_2, denotado por $KIJ(L_1, L_2)$, está dado por*

$$KIJ(L_1, L_2) = \frac{\delta(L_1) + \delta(L_2) - \delta(L_1 \cup L_2)}{\delta(L_1 \cup L_2)}.$$

Independizando el estudio de la similitud de secuencias del parámetro k. Además, realizando comparaciones con el coeficiente de Jaccard estándar, se muestra que KIJ es una buena medida para comparar similitud de secuencias. Sin embargo, los autores discuten que es necesario un estudio más profundo para ver cómo se compara KIJ con la distancia Mash que utiliza un valor de k en específico, pues al momento de calcular $KIJ(L_1, L_2)$ no necesariamente se utilizan los mismos valores de k para calcular $\delta(L_1), \delta(L_2)$ y $\delta(L_1 \cup L_2)$.

3.3. Reducir número de comparaciones

Estimar similitud utilizando herramientas como Dashing o Mash, conlleva comparar sketches entre sí. Menos comparaciones resultan en menos tiempo de ejecución, pero la precisión de los resultados se puede ver afectada. A continuación se presentan resultados con respecto a la cantidad de comparaciones. En particular, se discute el tamaño de los sketches de Hyperloglog para estimar un nuevo coeficiente de inclusión y también el cómo utilizar los sketches que se construyen con Dashing para omitir directamente ciertas comparaciones entre sketches.

3.3.1. Estimación del coeficiente de inclusión Φ

Dados dos conjuntos A, B, el coeficiente de inclusión entre A y B, denotado por $\Phi(A, B)$ se define como

$$\Phi(A, B) = \frac{|A \cap B|}{|A|}, \qquad A \neq \emptyset.$$

Es decir, $\Phi(A, B)$ corresponde a la fracción de elementos en A que también están en B. Este coeficiente es bastante similar al coeficiente de Jaccard, de hecho el coeficiente de Jaccard entre dos conjuntos X, Y corresponde al coeficiente de inclusión entre $X \cup Y$ e $X \cap Y$ (fracción de elementos en la unión que están en ambos conjuntos).

Sean A, B dos conjuntos y $S_p(A)$, $S_p(B)$ sus respectivos sketches de Hyperloglog, ambos de tamaño $m = 2^p$. En [14] se demuestra que $P = Pr[S_1(A) \leq S_1(B)]$ es función creciente de $\Phi(A, B)$, donde $S_1(A)$ y $S_1(B)$ son sketches de Hyperloglog de tamaño 1 de A y B, respectivamente. Así, se pueden usar los sketches $S_p(A)$ y $S_p(B)$ para obtener una estimación $\hat{P}$ de P. También se demuestra en [14] el siguiente teorema sobre el error de estimación de P:

Teorema 3.1. *La probabilidad de que el error de estimación de $\hat{P}$ sea a lo más e_P es*

$$Pr\left(|P - \hat{P}| \leq e_P\right) \geq 1 - 2\exp(-2^{m+1}e_p^2).$$

Así, el tamaño m nos permite ajustar el error de estimación e_P deseado. Finalmente, como P es función creciente de $\Phi(A, B)$, utilizando el método de la bisección es

posible obtener una aproximación $\hat{\Phi}$ de $\Phi(A, B)$ para la cual se obtiene el valor de $\hat{P}$.

Además del error numérico de la aproximación, también hay un error en la estimación de $\Phi(A, B)$ por usar $\hat{P}$ en lugar de P. En [14] se afirma que la gráfica de P como función de $\Phi(A, B)$ muestra que en un punto α se tiene que

$$\frac{e_\Phi}{e_P} = P'(\alpha).$$

Siendo e_Φ y e_P los errores de las estimaciones $\hat{\Phi}$ y $\hat{P}$, respectivamente. Así, a través del error e_P y de la derivada se controla el error e_Φ.

3.3.2. Criterio directo

Sean A, B dos conjuntos tales que $|A| \leq |B|$ y $h \in (0, 1)$. En [17] se demostró que

$$|A| < h|B| \Longrightarrow J(A, B) < h. \tag{3.3}$$

Así, si dentro de una base de datos de secuencias genómicas buscamos sólo los pares que tienen un coeficiente de Jaccard asociado mayor a un umbral h, la implicancia anterior permite establecer un criterio para descartar los pares que no cumplan la condición anterior. Sin embargo, si no se cumple la hipótesis de (3.3), no es posible asegurar que el coeficiente de Jaccard supere el umbral. A este criterio de selección le llamaremos **criterio directo** y ha sido utilizado previamente [17]. Formalmente, este criterio se puede definir como sigue.

Definición 3.3 (Criterio directo)**.** *Sea $k \in \mathbb{N}$ y sean $L_1, L_2 \in \Sigma^*$ dos secuencias genómicas tales que $d_k(L_1) \geq d_k(L_2)$. Entonces,*

- *Si $d_k(L_2) < h \cdot d_k(L_1)$, entonces el par de secuencias no se selecciona (se descarta).*
- *Si $d_k(L_2) \geq h \cdot d_k(L_1)$, entonces el par de secuencias sí se selecciona.*

Se destaca que este criterio no tiene margen de error en el descarte de un par: si un par de secuencias se descarta, es porque el coeficiente de Jaccard asociado es menor

que h. Sin embargo tiene una importante debilidad: sólo compara qué tan similares son los tamaños de los respectivos conjuntos, sin importar la información compartida entre ellos. Así, si en una base de datos se tienen conjuntos disjuntos de tamaños suficientemente similares (de forma que se cumpla la hipótesis de (3.3)), el criterio directo seleccionaría todos los pares, a pesar de que a ningún par le corresponda un Jaccard superior a h. Por otro lado, si todos los conjuntos tienen un Jaccard superior a h, entonces el criterio es perfecto, pues seleccionará a todos los pares. Podemos concluir que no es posible estimar con anterioridad qué tan efectivo será el criterio al momento de descartar pares de una base de datos.

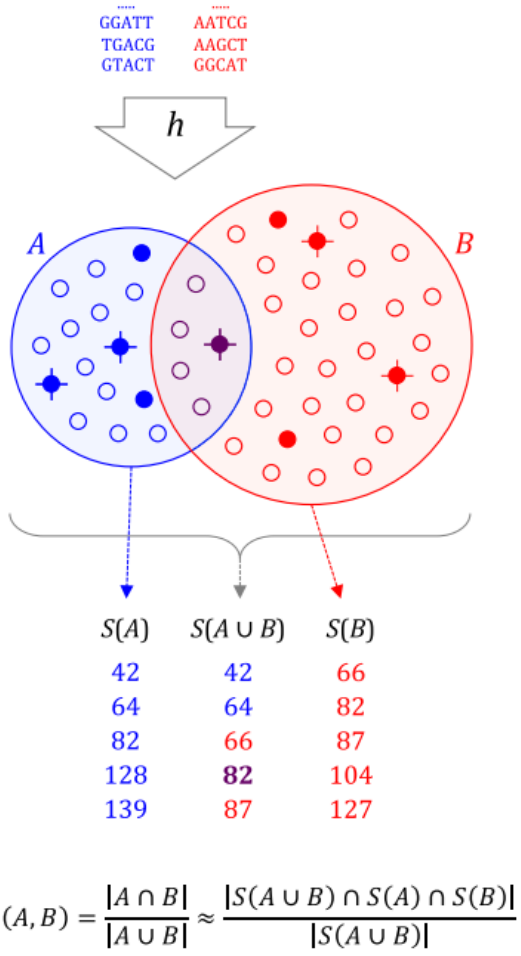

Figura 3.1: Resumen de la estrategia de MinsHash para estimar J. Primero, cada k-mer de los respectivos multiconjuntos de k-mers de dos secuencias es procesado por una función de hash h de 32 o 64 bits. Los conjuntos de hashes resultantes A y B contienen $|A|$ y $|B|$ hashes distintos (círculos pequeños). El coeficiente de Jaccard corresponde entonces a la fracción de hashes en común (círculos púrpuras). Este valor puede aproximarse utilizando una muestra aleatoria de $A \cup B$ mucho más pequeña. Los sketches de MinHash $S(A)$ y $S(B)$ de tamaño 5 asociados a A y B, respectivamente, se muestran en la figura. Estos son formados por los 5 menores valores de hash en cada conjunto (círculos rellenos). Utilizar $S(A)$ y $S(B)$ para obtener los 5 menores valores de $A \cup B$ (círculos con cruz), resulta en $S(A \cup B)$. Como $S(A \cup B)$ corresponde a una muestra aleatoria de $A \cup B$, la fracción de elementos en $S(A \cup B)$ que son compartidos tanto por $S(A)$ como por $S(B)$ es un estimados insesgado de $J(A, B)$. Fuente: [15]

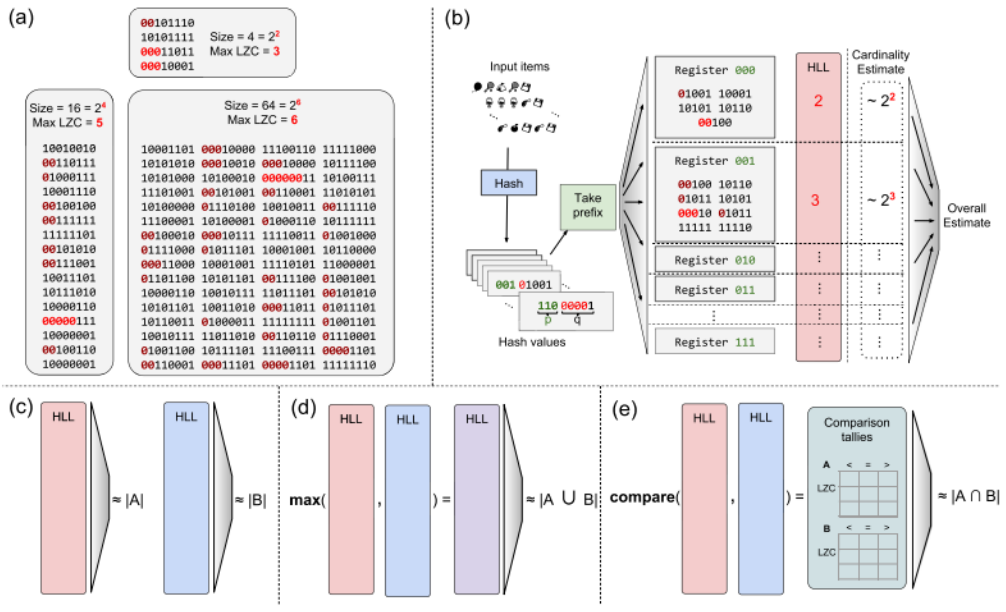

Figura 3.2: **(a)** Relación entre la máxima cantidad consecutiva de ceros a la izquierda (LMZ, Max Leading Zero Count) y la cardinalidad de conjuntos de números de 8 bits aleatoriamente generados. **(b)** Esquema de funcionamiento del sketch de Hyperloglog. **(c)** Estimación de la cardinalidad para los multiconjuntos A y B, y **(d)** estimación de la cardinalidad de su unión. Haciendo uso del principio de inclusión-exclusión es posible estimar la cardinalidad de la intersección. **(e)** Estimación directa de la cardinalidad de la unión utilizando el estimador JMLE de Ertl. Fuente: [1]

Capítulo 4

Construcción de nuevo criterio de selección

4.1. Definición del problema

Sea $h \in (0,1)$, $k, N \in \mathbb{N}$ y $\mathcal{M} = \{L_1, L_2, L_3, .., L_N\}$ un conjunto de secuencias genómicas. Supongamos que se obtienen todos los sketches de Hyperloglog de tamaño $m = 2^p$ de las secuencias en $\mathcal{M}$. Así, se tiene el conjunto de sketches:

$$S(\mathcal{M}) := \{S_p(L_i), \quad i \in \{1, 2, ..., N\}\},$$

donde $S_p(L)$ denota el sketch de Hyperloglog de tamaño $m = 2^p$ asociado a la secuencia L.

Se busca seleccionar el conjunto de pares de secuencias tales que el coeficiente de Jaccard correspondiente sea mayor que el umbral h. Es decir, se busca el conjunto dado por

$$\mathcal{S}(\mathcal{M}, k, h) := \{(S, S') \in \mathcal{M}^2 \ : \quad J_S(S, S', k) \geq h\}.$$

Utilizando los sketches en $S(\mathcal{M})$ se puede estimar J_S para todos los pares de secuencias. Sin embargo, en el proceso de cálculo de J_S es necesario estimar la cardinalidad de las uniones correspondientes. Gracias a la propiedad de fusión del Hyperloglog,

es posible estimar estas uniones comparando los respectivos sketches, sin embargo, esto requiere $N(N-1)/2$ comparaciones entre sketches de Hyperloglog de tamaño 2^p, dando un total de $2^{p-1}N(N-1)$ comparaciones entre números individuales. Por ejemplo, de acuerdo a Dashing [1], usando $p = 14$ se obtiene un error de estimación muy bajo. Luego, usando $p = 14$ el número total de comparaciones requeridas para el cómputo de todas las uniones es $2^{13}N(N-1)$.

El problema a resolver es utilizar los sketches individuales $S(\mathcal{M})$ para definir un criterio de selección que permita clasificar los pares de secuencias como pertenecientes o no pertenecientes a $\mathcal{S}(\mathcal{M}, k, h)$, con el fin de reducir la cantidad de comparaciones necesarias para encontrar $\mathcal{S}(\mathcal{M}, k, h)$.

Así, lo que se busca es reducir el espacio de búsqueda de $\mathcal{M}^2$ a algún subconjunto $\mathcal{M}^2$ de menor tamaño a través de un proceso barato en el sentido que el tiempo de ejecución total no supere al tiempo de ejecución resultante de no utilizar ningún criterio (realizar las $N(N-1)/2$ comparaciones de sketches). Por ejemplo, el criterio directo (Definición 3.3) permite descartar pares de forma muy barata para reducir el espacio de búsqueda, sin embargo ya vimos que este proceso tiene un problema importante al no utilizar nada de la información compartida entre secuencias.

4.2. Construcción de nuevo criterio hll_p de orden n

Dado que el objetivo es seleccionar los pares de secuencias de modo que el coeficiente de Jaccard J_S asociado sea mayor a un umbral, no necesitamos conocer el valor exacto de J_S, basta con una aproximación suficientemente buena. Por ejemplo, suponiendo $h = 0.8$, si se obtiene una estimación de 0.4 para J_S, y se puede asegurar con cierta probabilidad que esta estimación tiene un error no mayor a 0.3, entonces podríamos realizar el descarte de este par a pesar de estar teniendo un error de casi el 100 % en la estimación.

Por esta razón, estimar J_S de una forma menos costosa y con garantías de error,

permitiría crear un nuevo criterio de selección. Una forma inmediata de obtener una estimación de J_S de manera más rápida, es utilizar sketches de Hyperloglog de tamaño menor al ya establecido para hacer las mediciones más precisas. A continuación se presenta la obtención y definición de este criterio, además del respectivo análisis de error.

4.2.1. Cota para el error de estimación del coeficiente de Jaccard

Sean A, B dos multiconjuntos no vacíos de cardinalidades fijas tales que $\gamma|A| = |B|$, con $\gamma \in [0, 1]$. Es decir, $|A| \geq |B|$. Sean además $t = |A \cup B|$ y $\hat{t}_p = |A \cup B|_p$ el tamaño de la unión entre A y B, y su estimación dada por un Hyperloglog de tamaño p, respectivamente. Según (2.3), se tiene que

$$\hat{t}_p \in I := [t(1 - Z_\alpha \sigma_p), t(1 + Z_\alpha \sigma_p)] \subset \mathbb{R}^+,$$

con una probabilidad de α.

De lo anterior se pueden deducir las siguientes desigualdades (asumiendo que $\hat{t}_p \in I$):

$$0 \leq |\hat{t}_p - t| \leq Z_\alpha \sigma_p t. \tag{4.1}$$

$$0 < \frac{\hat{t}_p}{1 + Z_\alpha \sigma_p} \leq t. \tag{4.2}$$

Además, utilizando (4.2) y recordando que $|A| \leq t$, se tiene que

$$\frac{|A|}{t} \leq 1, \qquad \text{y} \qquad \frac{|A|}{t} \leq \frac{|A|(1 + Z_\alpha \sigma_p)}{\hat{t}_p}.$$

Así, como $(1 + \gamma) > 0$,

$$\frac{(1+\gamma)|A|}{t} \leq (1+\gamma) \operatorname{mín} \left\{ 1, \frac{|A|(1 + Z_\alpha \sigma_p)}{\hat{t}_p} \right\}. \tag{4.3}$$

Ahora, consideremos la función $f_J : I \to \mathbb{R}$ dada por

$$f_J(x) = \frac{|A| + |B| - x}{x} = \frac{(1+\gamma)|A|}{x} - 1.$$

Notemos que $f_J(t) = J(A,B)$. Además, $f_J(\hat{t}_p)$ es una estimación de $J(A,B)$ que denotaremos por $\hat{J}_p(A,B)$.

Sea $n \in \mathbb{N}$, notemos que f_J es diferenciable n veces en I. En efecto,

$$\forall x \in I: \quad f_J^{(n)}(x) = (-1)^n \frac{(1+\gamma)|A|}{x^{n+1}}.$$

Luego, dado $x_0 \in I$, y aplicando el Teorema de Taylor, se tiene que

$$\forall x \in I: \quad f_J(x) = \sum_{i=0}^{n} \frac{f_J^{(i)}(x_0)}{i!}(x-x_0)^i + R_n(x).$$

Tomando $x = \hat{t}_p \in I$ y $x_0 = t \in I$, se tiene que

$$f_J(\hat{t}_p) = \sum_{i=0}^{n} \frac{f_J^{(i)}(t)}{i!}(\hat{t}_p - t)^i + R_n(\hat{t}_p).$$

Luego, separando el primer término de la sumatoria y recordando que $f_J(t) = J(A,B)$ y $f_J(\hat{t}_p) = \hat{J}_p(A,B)$, se tiene que:

$$\hat{J}_p(A,B) - J(A,B) \approx \sum_{i=1}^{n} (-1)^i \frac{(1+\gamma)|A|}{t^{i+1} i!}(\hat{t}_p - t)^i. \tag{4.4}$$

De esta forma, podemos acotar a $e_J^{(p)}(A,B) := |\hat{J}_p(A,B) - J(A,B)|$ como sigue:

$$\begin{aligned}
|\hat{J}_p(A,B) - J(A,B)| &\approx \left| \sum_{i=1}^{n} (-1)^i \frac{(1+\gamma)|A|}{t^{i+1} i!}(\hat{t}_p - t)^i \right| \\
&\leq \sum_{i=1}^{n} \left| (-1)^i \frac{(1+\gamma)|A|}{t^{i+1} i!}(\hat{t}_p - t)^i \right| \\
&= \sum_{i=1}^{n} \frac{(1+\gamma)|A|}{t^{i+1} i!} |\hat{t}_p - t|^i \\
&\leq \sum_{i=1}^{n} \frac{(1+\gamma)|A|}{t^{i+1} i!} t^i Z_\alpha^i \sigma_p^i \qquad \text{(Por (4.1))} \\
&= \frac{(1+\gamma)|A|}{t} \sum_{i=1}^{n} \frac{(Z_\alpha \sigma_p)^i}{i!}
\end{aligned}$$

Finalmente, utilizando (4.3), se tiene que

$$|\hat{J}_p(A,B) - J(A,B)| \leq (1+\gamma)\, \text{mín} \left\{ 1, \frac{|A|(1+Z_\alpha \sigma_p)}{\hat{t}_p} \right\} \sum_{i=1}^{n} \frac{(Z_\alpha \sigma_p)^i}{i!}.$$

Utilizar el valor absoluto se justifica por la desigualdad triangular inversa (DTI). En efecto, si M_n denota a la sumatoria del miembro derecho de (4.4), entonces

$$\begin{aligned}\Big||\hat{J}_p(A,B) - J(A,B)| - |M_n|\Big| &= \Big||\hat{J}_p(A,B) - J(A,B)| - |-M_n|\Big| \\ &\leq \Big|\Big(\hat{J}_p(A,B) - J(A,B)\Big) + (-M_n)\Big| \qquad \text{(DTI)} \\ &= |R_n(\hat{t}_p)|.\end{aligned}$$

Así, si $\hat{J}_p(A,B) - J(A,B)$ es cercano a M_n, entonces $|\hat{J}_p(A,B) - J(A,B)|$ es igual de cercano a $|M_n|$ (o más cercano).

Luego, definiendo

$$C_{p,\alpha,n}(A,B) := (1+\gamma)\,\text{mín}\left\{1, \frac{|A|(1+Z_\alpha\sigma_p)}{\hat{t}_p}\right\}\sum_{i=1}^{n}\frac{(Z_\alpha\sigma_p)^i}{i!},$$

se tiene que

$$e_J^{(p)}(A,B) \leq C_{p,\alpha,n}(A,B).$$

Hemos encontrado entonces una cota para $e_J^{(p)}(A,B)$. Esta cota nos dice que la distancia entre $J(A,B)$ y $\hat{J}_p(A,B)$ es a lo más de $C_{p,\alpha,n}(A,B)$. Además,

$$C_{p,\alpha,n}(A,B) \leq (1+\gamma)\sum_{i=1}^{n}\frac{(Z_\alpha\sigma_p)^i}{i!},$$

obteniéndose una cota independiente de la estimación $\hat{t}_p$. Esta puede ser útil para calibrar el tamaño del Hyperloglog a usar antes de realizar las comparaciones.

Es importante que $\hat{t}_p$ pertenezca a I para que así las desigualdades se cumplan. Por esta razón se utilizará $\alpha = 95\,\%$ a menos que se especifique lo contrario. De esta forma $Z_\alpha = 1.96$. También sería válido tomar $Z_\alpha = 3$ para tener una confianza del $99.73\,\%$, sin embargo es necesario que Z_α sea lo menor posible para que $C_{p,\alpha,n}$ sea lo más ajustada posible.

A modo de ilustrar como se comporta $C_{p,\alpha,n}$, en la Figura 4.1 se muestra los valores que puede tomar considerando $n = 1$ y tomando dos casos extremos: $\gamma = 0$ y $\gamma = 1$ ($\gamma = 1$ indica multiconjuntos del mismo tamaño y $\gamma = 0$, que ambos multiconjuntos son de tamaño 0). En ambos casos se consideraron distintos valores de p. En los

gráficos se muestra cómo crece $C_{p,\alpha,n}$ en función de $|A|/\hat{t}_p$, este último valor se mueve entre 0.5 y 1 (pues se espera que $|A| \leq \hat{t}_p \leq 2|A|$). Se observa que al aumentar el valor de p, rápidamente la cota disminuye. Además, se distingue claramente cuando las dos posibles formas de $C_{p,\alpha,1}$, según el valor del mínimo que aparece en su expresión.

Se puede ver también en la Figura 4.1 que tomando $p = 6$, la cota no supera el valor de 0.5. Si bien un valor 0.5 puede ser grande (el coeficiente de Jaccard se mueve entre 0 y 1), puede ser un gran aporte cuando una pequeña parte de los multiconjuntos de k-mers son similares y la gran mayoría son poco similares (con respecto al coeficiente de Jaccard), tal como se comprobará en la sección 5.

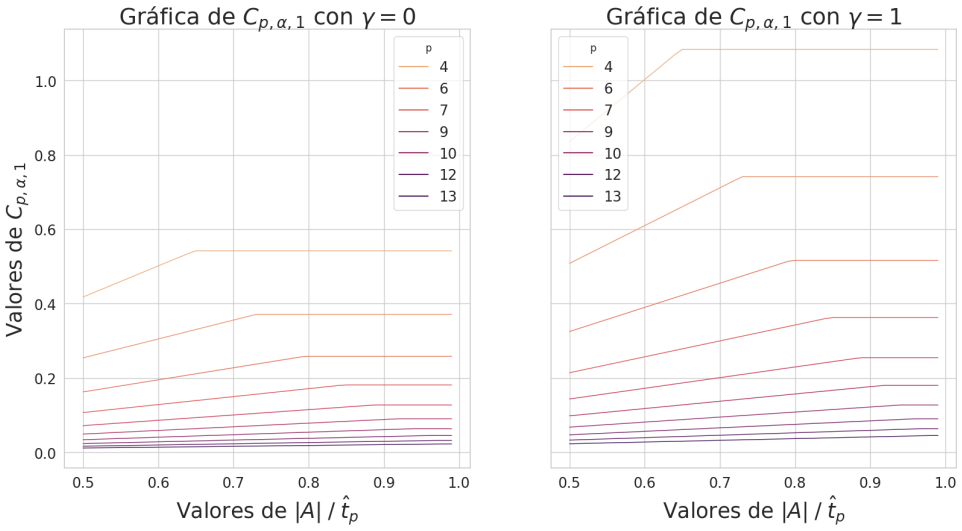

Figura 4.1: Gráfico de $C_{p,\alpha,1}$. A la izquierda se muestra el caso extremo $\gamma = 0$, y a la derecha, el caso extremo $\gamma = 1$. En ambos casos se muestra el valor de $C_{p,\alpha,1}$ para distintos valores de p y para valor de $|A|/\hat{t}_p$ entre 0.5 y 1. Ambas gráficas comparten el mismo eje vertical.

4.2.2. Análisis de error de aproximación $R_n(\hat{t}_p)$

La aproximación (4.4) tiene un error asociado $R_n(\hat{t}_p)$ que está dado por el Teorema de Taylor. Primero notemos que

$$\begin{aligned} |f_J^{(n+1)}(\hat{t}_p)| &= \frac{(1+\gamma)|A|}{\hat{t}_p^{n+2}} \\ &\leq \frac{(1+\gamma)|A|}{t^{n+2}(1-Z_\alpha\sigma_p)^{n+2}}. \end{aligned}$$

Así, según (2.2), se deduce que

$$\begin{aligned} |R_n(\hat{t}_p)| &\leq \frac{(1+\gamma)|A|}{t^{n+2}(1-Z_\alpha\sigma_p)^{n+2}} \, \frac{|\hat{t}_p - t|^{n+1}}{(n+1)!} \\ &\leq \frac{(1+\gamma)|A|t^{n+1}(Z_\alpha\sigma_p)^{n+1}}{t^{n+2}(1-Z_\alpha\sigma_p)^{n+2}(n+1)!} \qquad \text{(Por 4.1)} \\ &\leq \frac{2}{(1-Z_\alpha\sigma_p)(n+1)!}\left(\frac{Z_\alpha\sigma_p}{1-Z_\alpha\sigma_p}\right)^{n+1}. \end{aligned}$$

La última desigualdad se explica con que $(1+\gamma) \leq 2$ y $|A| \leq t$. Notar que si $Z_\alpha\sigma_p \leq 1/2$, entonces $\left(\frac{Z_\alpha\sigma_p}{1-Z_\alpha\sigma_p}\right) \leq 1$, de modo que aumentar el orden disminuiría el error. A modo de ejemplo, si tomamos $p = 6$ se tiene que $|R_1(\hat{t}_p)| \leq 0.1634$.

4.2.3. Criterio hll$_p$ de orden n

Utilizando la misma notación definida en la sub-sección anterior, notemos que

$$|J(A,B) - \hat{J}_p(A,B)| \leq C_{p,\alpha,n} \Longrightarrow J(A,B) - \hat{J}_p(A,B) \leq C_{p,\alpha,n}.$$

Así, $J(A,B) \leq \hat{J}_p(A,B) + C_{p,\alpha,n}$, por lo que

$$\hat{J}_p(A,B) + C_{p,\alpha,n} < h \Longrightarrow J(A,B) < h. \tag{4.5}$$

Esto nos permite construir el siguiente criterio de selección:

Definición 4.1 (Criterio hll$_p$ de orden n)**.** *Sean $p, n \in \mathbb{N}$ y sean $L_1, L_2 \in \Sigma^*$ dos secuencias genómicas. Entonces, tomando $A = spec_k(L_1)$ y $B = spec_k(L_2)$, y siguiendo la notación de la sub-sección anterior:*

- *Si* $\hat{J}_p(A,B) + C_{p,\alpha,n} < h$, *entonces el par no se selecciona (se descarta).*
- *Si* $\hat{J}_p(A,B) + C_{p,\alpha,n} \geq h$, *entonces el par sí se selecciona (se conserva).*

En la Figura 4.2 se muestra un esquema que muestra el funcionamiento del criterio propuesto.

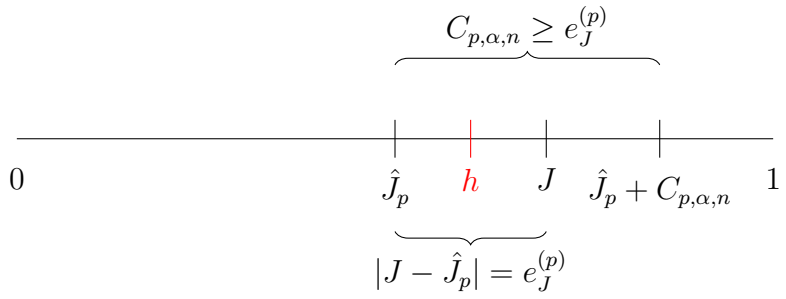

Figura 4.2: Esquema nuevo criterio: $C_{p,\alpha,n}$ permite construir un intervalo centrado en $\hat{J}_p$ donde se espera que se encuentre J. Así, si el extremo derecho de este intervalo $(\hat{J}_p + C_{p,\alpha,n})$ es menor que h, entonces se espera que J también sea menor que h.

Es claro entonces que el nuevo criterio depende principalmente de dos cosas. Primero, que la cota propuesta $C_{p,\alpha,n}$ se respete. De esta forma podemos asegurar que $J \leq \hat{J}_p + C_{p,\alpha,n}$ evitando que hayan falsos negativos. Lo segundo es que la cota propuesta $C_{p,\alpha,n}$ sea lo más pequeña posible, de esta forma $\hat{J}_p + C_{p,\alpha,n}$ está mucho más cerca de J, resultando en menos falsos positivos.

Capítulo 5

Resultados

5.1. Configuración de los experimentos

Los experimentos se implementaron en C++14 y se ejecutaron en un servidor con sistema operativo Linux del departamento de Ingeniería Eléctrica de la Universidad de Concepción. Este cuenta con un procesador Intel i9-10980XE con 18 núcleos físicos y 128 GB de RAM. Además, el servidor cuenta con un disco SSD de 1TB NVMe Class.

Para compilar se utilizó `g++` (GCC) versión 12.2.1. con las opciones `-std=c++14, -fopenmp, -O3, -match=native`. Como indica la segunda opción, se utilizó fopenmp para paralelizar el programa.

Para la implementación desarrollada se usó la biblioteca ofrecida por Dashing [1] para procesar las secuencias y construir los sketches de Hyperloglog. El código fuente está disponible en https://github.com/Zekess/criterio_seleccion

5.1.1. Valores de referencia

Para estudiar la efectividad del criterio propuesto se utilizaron los conjuntos de datos de RefSeq (ver Cuadro 5.1). Uno de los conjuntos de datos, denominado *prefs* en la Tabla 5.1, contiene 286 genomas y fue construido por los autores de Dashing [1]

con la finalidad de que los coeficientes de Jaccard de los respectivos pares cubran el rango [0,1].

El tamaño en memoria de las distintas secuencias es proporcional al largo de la secuencia misma. Vemos entonces que los distintos conjuntos abarcan variados rangos de tamaño. Desde secuencias entre 0.4KB y 2.4MB (conjunto *viral*), hasta secuencias entre 129MB y 4.2GB (conjunto *vertebrate_mammalian*). De esta forma, también se evalúa cómo se comportan los distintos criterios para distintos tamaños de secuencias.

Nombre	Símbolo	# elementos	# pares	Tamaño min.	Tamaño max.
prefs	$\mathcal{U}_1$	281	39340	5.4MB	12MB
bacteria50k	$\mathcal{U}_2$	50000	1249975000	111KB	4.1MB
archaea	$\mathcal{U}_3$	692	239086	480KB	5.6MB
fungi	$\mathcal{U}_4$	313	48828	2.2MB	133MB
plant	$\mathcal{U}_5$	145	10440	13MB	11GB
protozoa	$\mathcal{U}_6$	92	4186	6.3MB	222MB
vertebrate mammalian	$\mathcal{U}_7$	187	17391	129MB	4.2GB
vertebrate other	$\mathcal{U}_8$	271	36585	69MB	5.1GB
viral	$\mathcal{U}_9$	11232	63073296	0.4KB	2.4MB

Cuadro 5.1: Detalle de los conjuntos con los que se realizó la experimentación. En particular, *prefs* es un conjunto de secuencias de orígenes variados.

Por cada conjunto, se calcularon los sketches de Hyperloglogs necesarios y, utilizando la operación de fusión, se obtuvieron los sketches de Hyperloglog de las uniones de todos los pares posibles. Por cada par de secuencias S_1, S_2 que se procesa, siendo $X_1 = spec_k(S_1)$ y $X_2 = spec_k(S_2)$ y tomando a X_1 como el multiconjunto de mayor cardinalidad del par, se extraen las siguientes cantidades:

1. $\hat{J}_p(X_1, X_2)$: Estimación de $J(X_1, X_2)$.
2. $\hat{t}_p$: Estimación de $t = |X \cup Y|$ dada por los sketches auxiliares de tamaño p.
3. $e_J^{(p)}$: Error de estimación de J, es decir, $e_J^{(p)} = |J(X_1, X_2) - \hat{J}_p(X_1, X_2)|$.

Los valores de p utilizados fueron $4, 5, 6, 7, 8, 9, 10, 11, 12$ y 13. Además se utilizó

$k = 31$ (estándar de Dashing). Mientras no se especifique lo contrario, los valores de h y α son 0.8 y 0.95, respectivamente. Se tomaron los valores que se obtienen con un Hyperloglog de tamaño $p = 14$ como referencia (valores verdaderos), pues el error estándar relativo entregado por este Hyperloglog es de menos del 1 %. Así,

1. $J(X_1, X_2) = \hat{J}_{14}(X_1, X_2)$.
2. $|X_1| = |X_1|_{14}$.
3. $t = \hat{t}_{14}$.
4. $\gamma = \dfrac{|X_2|_{14}}{|X_1|_{14}}$.

5.2. Métricas de evaluación

El problema consiste en seleccionar los pares de secuencias cuyos coeficientes de Jaccard asociados sean mayor a un umbral. Este es un problema de clasificación. Así, con el fin de evaluar el criterio propuesto, se hacen las siguientes definiciones respecto a los pares de secuencias:

- **Verdadero Positivo o True Positive (TP)**: Par cuyo coeficiente de Jaccard asociado es mayor o igual que h y no es descartado por el criterio.
- **Falso Positivo o False Positive (FP)**: Par cuyo coeficiente de Jaccard asociado es menor que h y no es descartado por el criterio.
- **Verdadero Negativo o True Negative (TN)**: Par cuyo coeficiente de Jaccard asociado es menor que h y es descartado por el criterio.
- **Falso Negativo o False Negative (FN)**: Par cuyo coeficiente de Jaccard asociado es mayor o igual que h y es descartado por el criterio.

Para evaluar los resultados se utilizarán las siguiente métricas:

- **Precisión o Accuracy (ACC)**: Proporción entre pares clasificados correctamente y el total de pares procesados. Se calcula como

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}.$$

- **Sensibilidad o True Positive Rate (TPR)**: Capacidad del criterio de clasificar correctamente los pares positivos. Se calcula como

$$TPR = \frac{TP}{TP + FN}.$$

- **Especificidad o True Negative Rate (TNR)**: Capacidad del criterio de clasificar correctamente los casos negativos. Se calcula como

$$TNR = \frac{TN}{TN + FP}.$$

Notar que estas definiciones son válidas tanto para el criterio propuesto como para el criterio directo.

Además, también es necesario comparar las velocidades o tiempos de ejecución al utilizar el criterio. Para ello se considera la aceleración. La aceleración de un experimento A con respecto a un experimento B está dada por:

$$\text{Aceleración} = \frac{\text{Tiempo experimento } B}{\text{Tiempo experimento } A}.$$

Así, una aceleración mayor a 1 indica que el experimento A tarda menos tiempo que el experimento B (es más rápido).

Respecto al proceso de cálculo del coeficiente de Jaccard y de selección de pares, el mayor peso en cuanto a cálculos está en el número de comparaciones realizadas para estimar la cardinalidad de la unión. Supongamos que se tienen n secuencias genómicas y $N = n(n-1)/2$ pares de genomas. Entonces, si no se aplica ningún criterio, se deben realizar $2^{14}N$ comparaciones. De forma similar, utilizando el criterio de selección hll_p de orden n, se obtiene un número de comparaciones dado por

$$2^{14}(\text{FP} + \text{TP}) + 2^p \cdot N.$$

Pues se deben comparar todos los pares clasificados como positivos por un Hyperloglog de tamaño 14 y además se hacen todas las comparaciones entres los sketches de tamaño p. Se espera que el número de comparaciones sea alto para valores pequeños

de p como $p = 4$ o $p = 5$, pues el error resultante del Hyperloglog es más grande, generando una mayor cantidad de falsos positivos. También se espera que para valores grandes de p, como $p = 12$ o $p = 13$, el número de comparaciones también sea alto, pues en este caso los sketches auxiliares son casi del mismo tamaño que el sketch grande que es de tamaño 14, provocando que el cálculo auxiliar sea casi tan pesado como el cálculo principal (aunque el número de falsos positivos debería ser mucho más pequeño).

Por otro lado, el criterio directo resulta en un número de comparaciones dado por

$$2^{14}(\mathrm{FP} + \mathrm{TP}),$$

ya que sólo compara los sketches de Hyperloglog de tamaño 14 de los pares clasificados como positivos.

Se espera que el criterio hll_p tenga una menor cantidad de falsos positivos (FP) que el criterio directo, de manera que se reduzca el número de comparaciones. En cuanto a los verdaderos positivos (TP), no hay competencia realmente, pues el criterio directo detecta a todos los verdaderos positivos de forma exacta, su problema es la cantidad de falsos positivos.

Respecto al uso de memoria, es claro que el criterio directo no ocupa nada de memoria adicional, a diferencia del criterio hll_p que sí utiliza los sketches auxiliares (aunque estos son pequeños). Para medir este impacto, se considerará la memoria utilizada por cada criterio como la cantidad de buckets totales que se deben almacenar. Así, el no utilizar ningún criterio resulta en un uso de memoria de $2^{14} \cdot n$ unidades. Utilizar el criterio directo, como ya se mencionó, resulta en la misma cantidad de uso de memoria, pues no se utiliza memoria adicional. Sin embargo, el criterio propuesto, considerando un Hyperloglog de tamaño p, utiliza una cantidad de memoria dada por

$$(2^{14} + 2^p) \cdot n.$$

Es decir, este criterio ocupa $(1 + 2^{p-14})$ veces la memoria que utiliza el proceso sin ningún criterio de selección. Si normalizamos con respecto a la memoria que usa el caso base (sin criterio), se obtiene la gráfica de la Figura 5.1.

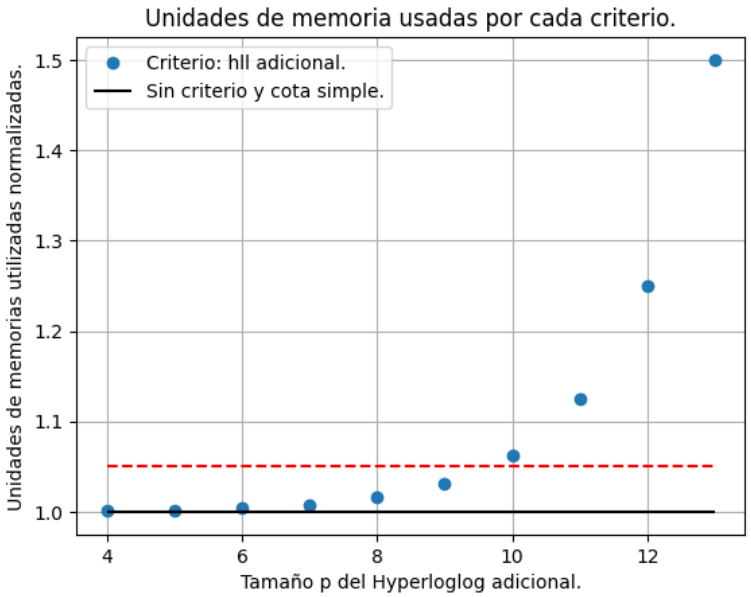

Figura 5.1: Fracción de memoria usada por cada criterio, con respecto a la memoria utilizada al no usar ningún criterio. La línea roja es para referencia y representa el 5 % de memoria adicional.

Se observa que hasta $p = 10$, la memoria adicional es menos del 10 %. Incluso, utilizando valores de p menores que 10, nos aseguramos un uso adicional de memoria menor al 5 %.

5.3. Criterio hll_p de orden 1

Una cota $C_{p,\alpha,n}$ más pequeña implica una menor cantidad de falsos positivos. En otras palabras, se seleccionarán menos pares que tengan un coeficiente de Jaccard menor al umbral h, lo que resultará en un menor tiempo invertido en comparar sketches de Hyperloglog de gran tamaño. Por lo tanto, es de nuestro interés elegir los parámetros de manera que la cota sea lo más pequeña posible.

El parámetro p regula el tamaño de los sketches auxiliares. Buscamos que p sea lo suficientemente pequeño como para que la construcción de los sketches de Hyperloglog auxiliares y el cálculo de $C_{p,\alpha,n}$ no sean costosos (justificando así el uso del

criterio). Sin embargo, también es necesario que p sea lo suficientemente grande para que el error asociado con los sketches de Hyperloglog auxiliares sea bajo y, por ende, la cota $C_{p,\alpha,n}$ no sea muy holgada.

El parámetro α regula la confianza en la estimación proporcionada por los sketches auxiliares. Aunque $C_{p,\alpha,n}$ aumenta conforme aumenta α, la construcción de esta cota depende de que se satisfagan las desigualdades (4.1) y (4.2). En consecuencia, como se mencionó en la sección 4.2.1, se eligió α como 0.95.

Finalmente, el parámetro n controla el error de la aproximación del Teorema de Taylor. Valores más grandes de n implican una mejor aproximación, pero también una cota $C_{p,\alpha,n}$ más grande, mientras que valores más pequeños de n resultan en lo contrario. Por lo tanto, es conveniente seleccionar $n = 1$ si se busca minimizar la cota, pero existe el riesgo de que el error de aproximación sea mayor y, por lo tanto, haya más falsos negativos (disminuyendo el valor de TPR).

El objetivo del filtrado de pares de secuencias es reducir la cantidad de comparaciones costosas a realizar, pero minimizando la pérdida de pares cuyo coeficiente de Jaccard asociado sea mayor que h, pues esta es la información valiosa que se desea rescatar finalmente. Por lo tanto, es necesario encontrar un equilibrio con el valor de n de manera que sea lo más bajo posible y aún así logre un valor de TPR suficientemente alto.

A través de experimentos, se observó que el criterio de orden 1 ofrece buenos resultados en términos de TPR, alcanzando valores superiores al 0.99. Por esta razón, los resultados experimentales corresponden principalmente a los del criterio de orden 1.

5.3.1. Éxito de la cota $C_{p,\alpha,1}$

Como primer resultado, se evalúa si la cota propuesta efectivamente se respeta. Para ello, se calcula, por cada clase, el porcentaje de pares para los cuales $e_J^{(p)} \leq C_{p,\alpha,n}$. Los resultados se presentan en el Cuadro 5.2.

La primera observación es que, tal como se esperaba, en todas las clases se obtuvo una tasa de éxito cercana al 95 % para la cota propuesta. Además, en promedio, el

Criterio	$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$	$\mathcal{U}_4$	$\mathcal{U}_5$	$\mathcal{U}_6$	$\mathcal{U}_7$	$\mathcal{U}_8$	$\mathcal{U}_9$	**Media**
hll_4	92.56	97.62	98.15	97.25	98.03	96.25	96.76	97.58	96.88	**96.79**
hll_5	97.55	97.44	97.95	97.28	97.84	96.97	94.38	97.81	96.87	**97.12**
hll_6	96.47	97.84	97.87	98.43	95.76	97.42	95.15	97.24	96.66	**96.98**
hll_7	96.35	97.93	96.98	98.09	96.21	97.99	95.88	98.19	96.80	**97.16**
hll_8	93.87	96.79	97.67	97.54	97.22	97.61	95.91	97.29	96.73	**96.74**
hll_9	93.77	93.81	96.80	95.87	93.96	96.82	94.43	97.63	96.57	**95.52**
hll_{10}	94.45	94.95	95.42	96.51	95.30	97.87	96.53	95.59	96.45	**95.90**
hll_{11}	96.80	95.88	97.19	96.05	96.23	97.64	97.06	96.89	97.19	**96.77**
hll_{12}	95.03	94.93	97.61	97.81	96.55	97.25	97.88	97.50	98.00	**96.95**
hll_{13}	98.73	98.88	99.19	98.87	99.20	99.57	98.64	98.82	99.41	**99.03**
Media	**95.56**	**96.61**	**97.48**	**97.37**	**96.63**	**97.54**	**96.26**	**97.45**	**97.16**	**96.90**

Cuadro 5.2: Porcentaje de éxito de la cota $C_{p,\alpha,1}$

.

éxito se mueve entre 95.56 % y 97.54 % entre clases y entre 95.52 % y 99.03 % para distintos valores de p.

5.3.2. Métricas de evaluación para $\alpha = 0.95$ y $h = 0.8$

Respecto a la métricas de evaluación ACC, TPR y TNR, los resultados se presentan en las Cuadros 5.3, 5.4 y 5.5.

Se observa que el ACC y el TNR se comportan muy similar. Esto se debe a que la gran proporción de pares son verdaderos negativos o falsos positivos. Como el TNR representa qué tan bien se filtran los pares negativos, es la clave para reducir los tiempos de comparación. En promedio, desde $p = 6$ ya se alcanza un TNR del 98.4 %, siendo una mejora considerable en comparación al 57.5 % y 88.6 % obtenidos con $p = 4$ y $p = 5$. Comparando con el criterio directo, vemos que desde $p = 5$ el criterio propuesto ya es mejor filtrando los pares negativos. Sin embargo, depende del conjunto qué tanta es esta diferencia. El filtrado dado por el criterio directo depende totalmente del conjunto, siendo factible fabricar un conjunto artificial donde el TNR de este criterio sea de 0. Mas, el TNR del criterio propuesto se muestra consistente a lo largo de los distintos tipos de conjuntos, además de mejorar al aumentar el valor

Criterio	$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$	$\mathcal{U}_4$	$\mathcal{U}_5$	$\mathcal{U}_6$	$\mathcal{U}_7$	$\mathcal{U}_8$	$\mathcal{U}_9$	**Media**
Directo	23.3	62.9	67.1	70.0	84.6	81.1	48.5	78.8	89.4	**67.3**
hll_4	58.8	52.5	48.2	53.2	64.1	52.2	63.0	58.6	67.2	**57.6**
hll_5	84.8	85.9	82.8	89.5	92.8	89.9	87.1	91.7	93.0	**88.6**
hll_6	93.6	96.8	98.6	99.6	99.6	99.6	98.7	99.5	99.8	**98.4**
hll_7	95.9	97.7	99.2	100.0	99.9	99.9	99.7	99.9	100.0	**99.1**
hll_8	97.8	99.1	99.2	100.0	100.0	100.0	99.8	99.9	100.0	**99.5**
hll_9	98.9	99.9	99.3	100.0	100.0	100.0	99.8	100.0	100.0	**99.8**
hll_{10}	98.9	99.9	99.3	100.0	100.0	100.0	99.9	100.0	100.0	**99.8**
hll_{11}	99.1	100.0	99.7	100.0	100.0	100.0	99.9	100.0	100.0	**99.8**
hll_{12}	99.5	100.0	99.7	100.0	100.0	100.0	99.9	100.0	100.0	**99.9**
hll_{13}	99.9	100.0	99.9	100.0	100.0	100.0	99.9	100.0	100.0	**100.0**
Media	**86.4**	**90.4**	**90.3**	**92.0**	**94.6**	**93.0**	**90.6**	**93.5**	**95.4**	**91.8**

Cuadro 5.3: Precisión del criterio de orden 1 sobre las distintas clases y para distintos valores de p.

Criterio	$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$	$\mathcal{U}_4$	$\mathcal{U}_5$	$\mathcal{U}_6$	$\mathcal{U}_7$	$\mathcal{U}_8$	$\mathcal{U}_9$	Media
Directo	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
hll_4	99.7	99.8	100.0	100.0	100.0	100.0	90.0	100.0	99.5	**98.8**
hll_5	100.0	99.7	100.0	100.0	100.0	80.0	100.0	100.0	99.0	**97.6**
hll_6	100.0	99.9	98.6	100.0	100.0	100.0	100.0	100.0	99.3	**99.8**
hll_7	100.0	99.9	100.0	100.0	100.0	100.0	100.0	100.0	99.8	**100.0**
hll_8	100.0	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
hll_9	99.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
hll_{10}	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
hll_{11}	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
hll_{12}	97.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**99.7**
hll_{13}	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
Media	**99.7**	**99.9**	**99.9**	**100.0**	**100.0**	**98.2**	**99.1**	**100.0**	**99.8**	**99.6**

Cuadro 5.4: TPR del criterio de orden 1 sobre las distintas clases y para distintos valores de p.

de p.

Criterio	$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$	$\mathcal{U}_4$	$\mathcal{U}_5$	$\mathcal{U}_6$	$\mathcal{U}_7$	$\mathcal{U}_8$	$\mathcal{U}_9$	Media
Directo	22.6	62.6	67.1	70.0	84.6	81.1	48.5	78.8	89.4	**67.2**
hll_4	58.5	52.3	48.1	53.2	64.1	52.2	63.0	58.6	67.2	**57.5**
hll_5	84.7	85.8	82.7	89.5	92.8	89.9	87.1	91.7	93.0	**88.6**
hll_6	93.5	96.8	98.6	99.6	99.6	99.6	98.7	99.5	99.8	**98.4**
hll_7	95.8	97.7	99.2	100.0	99.9	99.9	99.7	100.0	100.0	**99.1**
hll_8	97.8	99.1	99.2	100.0	100.0	100.0	99.8	100.0	100.0	**99.5**
hll_9	98.9	99.9	99.3	100.0	100.0	100.0	99.9	100.0	100.0	**99.8**
hll_{10}	98.9	99.9	99.3	100.0	100.0	100.0	99.9	100.0	100.0	**99.8**
hll_{11}	99.1	100.0	99.7	100.0	100.0	100.0	99.9	100.0	100.0	**99.8**
hll_{12}	99.6	100.0	99.7	100.0	100.0	100.0	100.0	100.0	100.0	**99.9**
hll_{13}	99.9	100.0	99.9	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
Media	**86.3**	**90.4**	**90.2**	**92.0**	**94.6**	**93.0**	**90.6**	**93.5**	**95.4**	**91.8**

Cuadro 5.5: TNR del criterio de orden 1 sobre las distintas clases y para distintos valores de p.

La sensibilidad (o TPR) con el criterio propuesto casi siempre es del 100 % o muy cercana. A excepción de dos casos, para $\mathcal{U}_6$ con $p = 5$ y para $\mathcal{U}_7$ con $p = 4$. Sin embargo, al observar con más atención estos casos, resulta que el total de positivos es de 5 y 10, respectivamente. Y el criterio se equivocó 1 vez en cada caso, bajando abruptamente el TPR.

El criterio clasifica casi perfectamente a todos los pares positivos. En promedio, desde $p = 6$ el TPR no baja del 99.7 %.

Dado que todos los pares positivos son clasificados casi perfectamente, la cantidad de falsos negativos FN es casi 0. De esta forma,

$$\begin{aligned} 1 - ACC &= 1 - \frac{TP + TN}{TP + FP + TN + FN} \\ &= \frac{FP + FN}{TP + FP + TN + FN} \\ &\approx \frac{FP}{TP + FP + TN + FN}. \end{aligned}$$

Es decir, gracias a que TPR es muy cercano al 100 %, se puede deducir que $1 - ACC$ es muy cercano a la proporción de FP con respecto al total de pares. Esta es la proporción que se desea reducir.

Por otro lado, también se realizaron las mediciones para otros valores de h y para otros valores de Z_α. Estos se muestran en las Cuadros 5.6 y 5.7. Se escogió utilizar un valor de $p = 6$ pues es el valor que mejor desempeño mostraba en los experimentos pasados. Las métricas se calcularon sobre el conjunto *bacterias50k*.

h	Criterio	ACC	TPR	TNR
0.7	Directo	0.4063	1	0.402394
0.8	Directo	0.628573	1	0.626404
0.9	Directo	0.789781	1	0.788908
0.7	hll_6	0.954202	0.991931	0.953953
0.8	hll_6	0.967977	0.999157	0.967795
0.9	hll_6	0.971854	0.997699	0.971747

Cuadro 5.6: Métricas para distintos valores de h ($\alpha = 0.95$)

α	Criterio	ACC	TPR	TNR
-	Directo	0.628573	1	0.626404
0.683	hll_6	0.983443	0.991594	0.983396
0.95	hll_6	0.967977	0.999157	0.967795
0.997	hll_6	0.874456	0.999977	0.873723

Cuadro 5.7: Métricas para distintos valores de α ($h = 0.8$)

Se observa que ambos criterios mejoran al aumentar el valor de h, pero el criterio directo mejora de forma mucho más notoria. Por otro lado, al aumentar α, el TPR del criterio hll_p aumenta, pero el TNR baja, pues la cota se vuelve más holgada. Aún así, el criterio propuesto se observa bastante consistente en los distintos casos. Si bien el TNR tiene sentido que sea más alto para $\alpha = 0.684$, llama la atención el resultado del TPR para este mismo caso, es muy alto. Es posible que la cota sea más holgada de lo que se esperaba,

5.3.3. Tiempos de comparación

Se realizó el experimento de obtener los pares positivos para cada clase, sin filtrar con algún criterio (realizando todas las comparaciones entre sketches de tamaño 14), utilizando el criterio directo y, el criterio hll_p de orden 1. Para las 3 situaciones se midió el tiempo de ejecución.

Los tiempos totales de cada experimento se muestran en las Figuras 5.2, 5.3 y 5.4. Los tiempos se muestran en escala logarítmica ya que la diferencia de tiempo es muy grande entre los conjuntos más pequeños y los más grandes. Se ve que en los tres casos, el tiempo que toma en realizar las comparaciones para seleccionar el conjunto que cumpla la condición $J > h$ se comporta de forma similar. Además, se observa que para las distintas clases, el criterio hll_p de orden 1 se comporta similar para los distintos valores de p. El menor tiempo de comparación se alcanza entre $p = 6$ y $p = 8$.

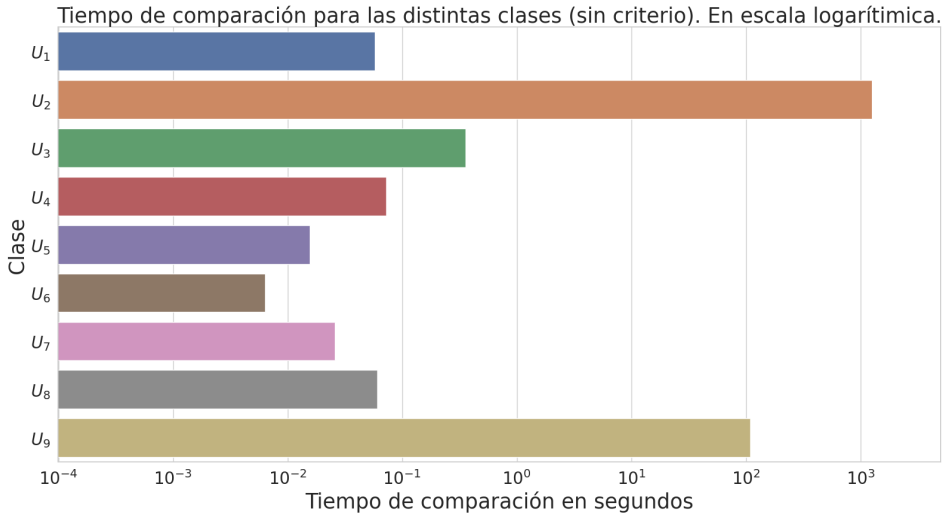

Figura 5.2: Gráfico del tiempo de comparaciones para las distintas clases al no usar ningún criterio para filtrar.

Por otro lado, para estudiar la ganancia, se evalúa la aceleración del experimento al utilizar el criterio hll_p de orden 1 con respecto a no utilizar ningún criterio de filtrado (Ver Figura 5.5).

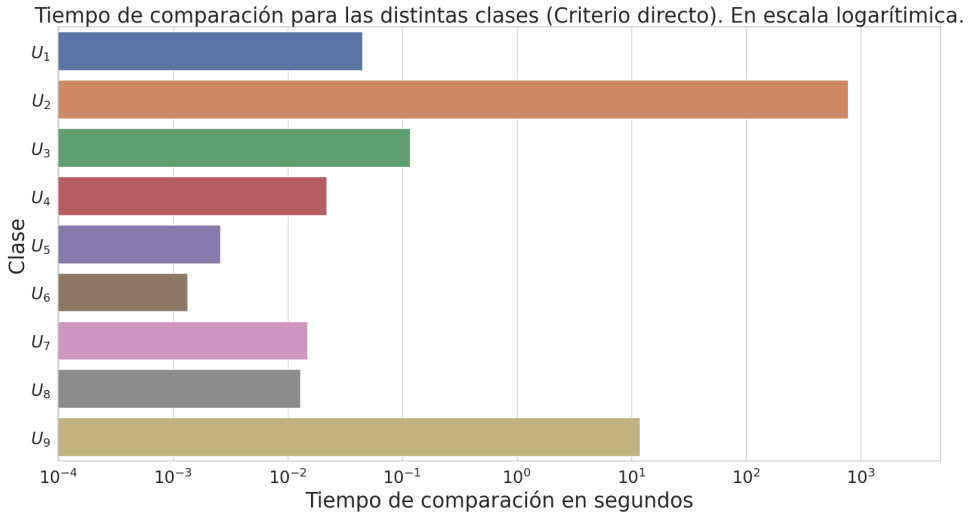

Figura 5.3: Gráfico del tiempo de comparaciones para las distintas clases al usar el criterio directo para filtrar.

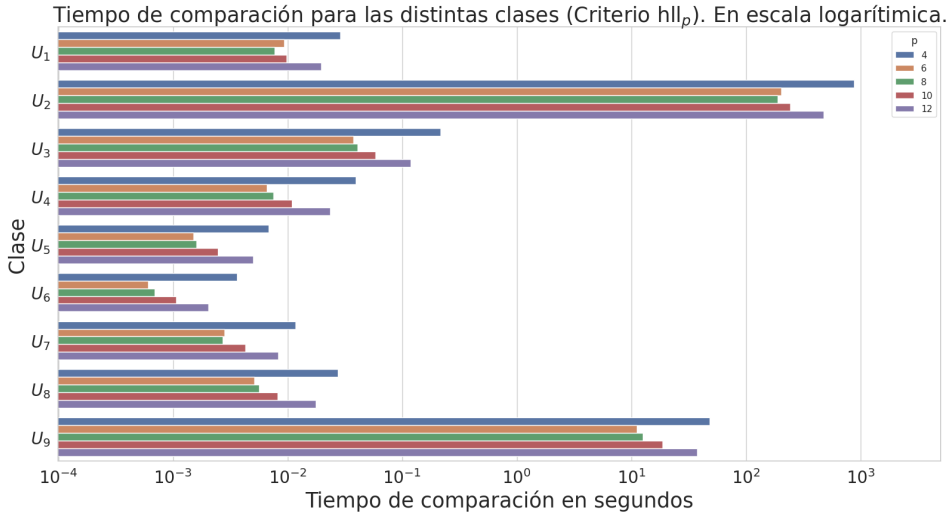

Figura 5.4: Gráfico del tiempo de comparaciones para las distintas clases al usar el criterio hll_p de orden 1 con distintos valores de p para filtrar.

Guiándonos por la media geométrica de las aceleraciones (media estándar para comparar aceleraciones) para las distintas clases, se observa que entre $p = 6$ y $p = 8$

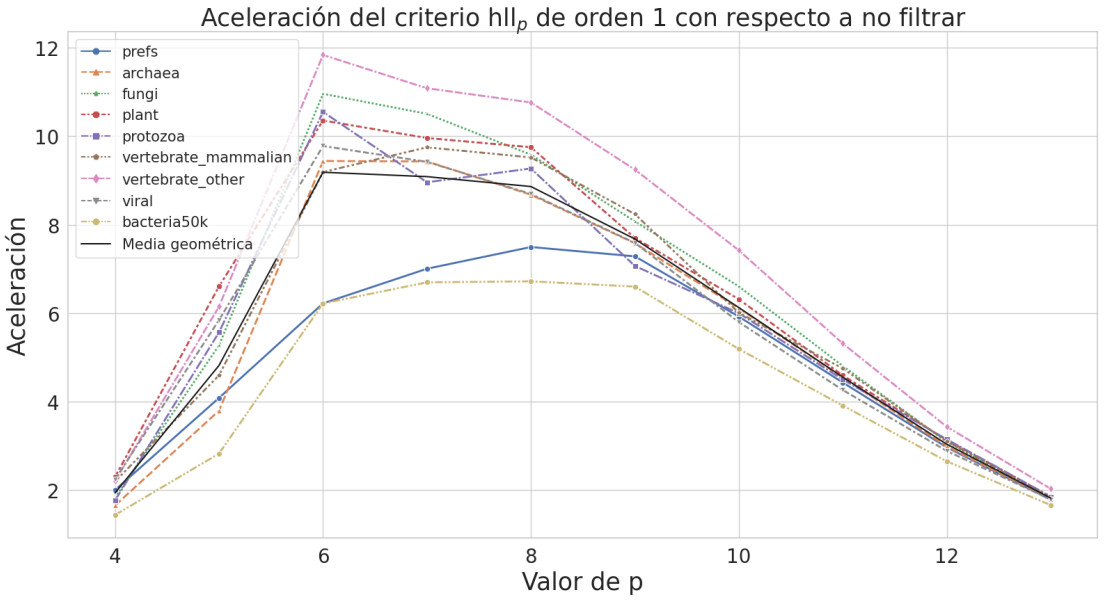

Figura 5.5: Gráfico de la aceleración del tiempo de comparaciones usando el criterio hll_p de orden 1 con respecto a no usar ningún proceso de filtrado, usando distintos tamaños p para los sketches auxiliares.

es cuando más ganancia hay en cuanto a tiempo, lográndose que las comparaciones se ejecuten cerca de 9 veces más rápido. Tal como se esperaba, para bajos valores de p la aceleración es más baja, pues el filtrado no es tan bueno (bajo TNR). Al otro extremo, cuando p es alto, el filtrado es excelente, pero se vuelve muy costoso calcular $C_{p,\alpha,1}$, al punto de no ser realmente conveniente aplicar el criterio, pues se comparan sketches auxiliares de tamaños muy cercanos a los sketches grandes.

Comparamos también la aceleración del criterio hll_p de orden 1 con respecto al criterio directo (Ver Figura 5.6). La media geométrica indica que entre $p = 6$ y $p = 8$ se alcanza una aceleración cercana a 2.5. Es decir, se reduce el tiempo de comparaciones a menos de la mitad. Sin embargo, hay más varianza entre las distintas clases, están más separadas las curvas. Esto se debe a que el criterio directo tiene variados TNR para las clases, y la ventaja en tiempo del criterio hll_p depende justamente de la ventaja en cuanto al TNR. Por esta razón, las dos curvas que alcanzan las mayores aceleraciones corresponden a las clases donde el criterio directo tiene los

TNR más bajos. De forma similar, la aceleración es más baja para las clases que tienen un TNR alto para el criterio directo.

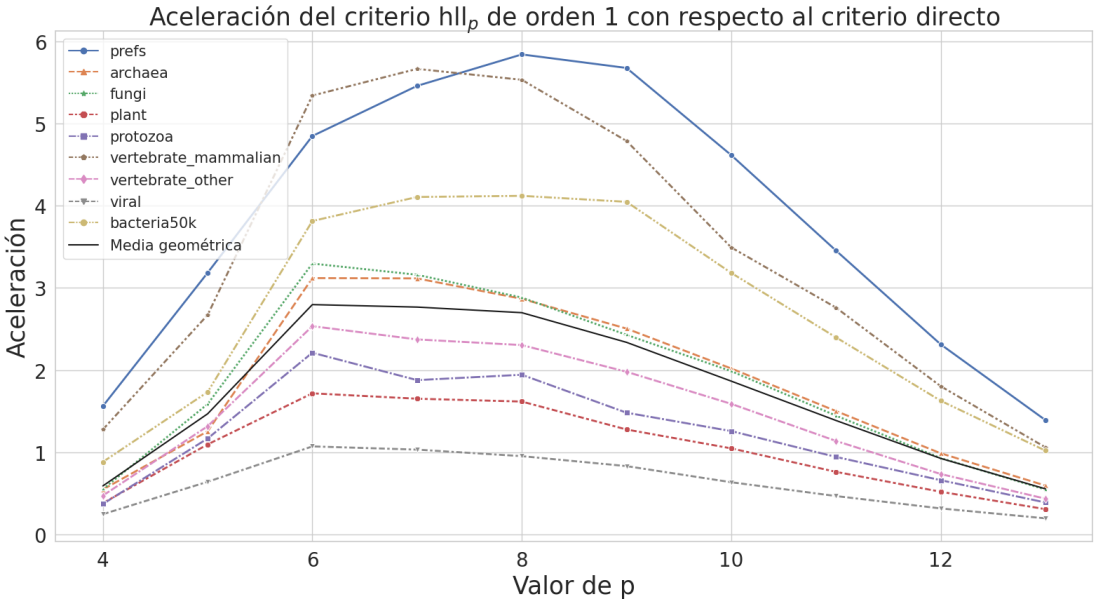

Figura 5.6: Gráfico de la aceleración del tiempo de comparaciones usando el criterio hll_p de orden 1 con respecto al usar el criterio directo, usando distintos tamaños p para los sketches auxiliares.

Una ventaja del criterio propuesto entonces, es la consistencia que presenta al momento de filtrar (como se ve en la Figura 5.5). A diferencia del criterio directo que depende muy fuertemente del conjunto de secuencias y de las cardinalidades de los respectivos conjuntos de k-mers.

Como se mencionó en la sección 3, es posible fabricar un conjunto para el cual el criterio directo tenga un TNR de 0, secuencias tales que sus conjuntos de k-mers tengan cardinalidades muy similares pero que sean disjuntos. En este caso el criterio directo se comporta igual a no utilizar ningún filtrado y realizar todas las comparaciones posibles. Sin embargo, como los Jaccard en este caso serían muy cercanos a 0, lo más probable es que el criterio hll_p no tenga problemas y clasifique a todos los pares como negativos, logrando un TNR cercano a 1. Por el contrario, también es posible fabricar un conjunto de secuencias de modo que los conjuntos de

k-mers tengan el mismo tamaño y todos los pares un coeficiente de Jaccard mayor al umbral h. En este caso, el criterio directo filtraría a la perfección, superando al criterio hll_p. Sin embargo, en este caso la realidad es que no se estaría haciendo ningún filtrado, pues todos los pares son positivos, por lo que efectivamente ambos criterios no ahorrarían tiempo en comparación a no utilizar ningún criterio. Así, el caso que favorece al criterio directo es un caso de poco interés, pues el problema original desaparece (filtrar lo más posible).

Similar a como se hizo con las métricas, también se realizaron mediciones de tiempo para otros valores de h y α. Estos se muestran en las Cuadros 5.8 y 5.9. Se escogió utilizar un valor de $p = 6$ pues es el valor que mejor desempeño mostraba en los experimentos pasados. Las trabajó sobre el conjunto *bacterias50k*.

h	Criterio	Construcción	Comparaciones
0.7	Directo	282.035	774.863
0.8	Directo	278.411	482.172
0.9	Directo	278.196	274.545
0.7	hll_6	349.435	203.197
0.8	hll_6	349.564	181.547
0.9	hll_6	349.461	181.336

Cuadro 5.8: Tiempos para distintos valores de h ($\alpha = 0.95$), en segundos.

α	Criterio	Construcción	Comparaciones
-	Directo	278.411	482.172
0.683	hll_6	349.539	159.947
0.95	hll_6	349.564	181.547
0.997	hll_6	349.383	316.789

Cuadro 5.9: Tiempos para distintos valores de α ($h = 0.8$), en segundos.

Los resultados son acordes a las métricas obtenidas para estos casos. En particular se observa la desventaja de usar un valor muy alto de α, pues la holgura de la cota $C_{p,\alpha,n}$ provoca muchos falsos positivos (bajo TNR), ralentizando el proceso de comparación. Por esta razón, pareciera que $\alpha = 0.95$ lleva a un equilibrio entre una cota más ajustada y que al mismo tiempo se cumpla.

5.4. Criterio hll_p de orden superior

A pesar de los buenos resultados del criterio hll_p de orden 1, es natural revisar qué sucede si se aumenta el orden. Dado que la cota $C_{p,\alpha,n}$ crece y el error disminuye, se espera que el TPR aumente y el TNR baje, es decir, se seleccionarán de mejor forma los pares positivos, pero se dejarán pasar muchos pares negativos.

Dado que el objetivo es ilustrar lo que sucede al aumentar el orden del criterio, sólo se trabajó con la clase *prefs*. Se utilizó $\alpha = 0.95$ y se realizó el experimento para distintos valores de p. Los resultados se muestran en Cuadro 5.10 y Cuadro 5.11

Criterio	Éxito $C_{p,\alpha,2}$	ACC	TPR	TNR
Directo	N.A.	0.233	1.000	0.226
hll_4	0.962	0.435	1.000	0.429
hll_5	0.989	0.772	1.000	0.770
hll_6	0.980	0.931	1.000	0.931
hll_7	0.978	0.953	1.000	0.953
hll_8	0.954	0.975	1.000	0.975
hll_9	0.956	0.989	1.000	0.989
hll_{10}	0.979	0.989	1.000	0.989
hll_{11}	0.987	0.991	1.000	0.991
hll_{12}	0.968	0.995	0.975	0.995
hll_{13}	0.997	0.999	1.000	0.999

Cuadro 5.10: Métricas al utilizar el criterio de orden 2 con el conjunto *prefs*, usando distintos valores de p. Se utilizó $\alpha = 0.95$ y $h = 0.8$.

Se observa que el criterio, como se esperaba, mejora el porcentaje de éxito de la cota $C_{p,\alpha,n}$ al aumentar n. Sin embargo, el aumento de la cota es bastante pequeño, al punto en que, si bien se ven mejores resultados, no son realmente significativos. Se respalda la idea de que el criterio de orden 1 es suficiente para realizar el filtrado. Sin embargo, el aumento del orden, y por lo tanto de la cota, tuvo un impacto pequeño en el TNR, por lo que igual puede considerarse usar un criterio de orden mayor, ya que el costo computacional es el mismo para cualquier orden y se tendrá una mayor confianza sobre la selección de pares positivos (TNR).

Criterio	Éxito $C_{p,\alpha,3}$	ACC	TPR	TNR
Directo	N.A.	0.233	1.000	0.226
hll_4	0.965	0.412	1.000	0.405
hll_5	0.990	0.760	1.000	0.758
hll_6	0.981	0.931	1.000	0.930
hll_7	0.979	0.953	1.000	0.953
hll_8	0.955	0.975	1.000	0.976
hll_9	0.956	0.989	1.000	0.989
hll_{10}	0.980	0.989	1.000	0.989
hll_{11}	0.987	0.991	1.000	0.991
hll_{12}	0.968	0.995	0.975	0.995
hll_{13}	0.997	0.999	1.000	0.999

Cuadro 5.11: Métricas al utilizar el criterio de orden 3 con el conjunto *prefs*, usando distintos valores de p. Se utilizó $\alpha = 0.95$ y $h = 0.8$.

5.5. Descarte de pares parecidos

Dado que el problema está en encontrar pares cuyo coeficiente de Jaccard supere un umbral, el criterio se construyó de forma de descartar aquellos pares que muy seguramente no cumplían la condición. Sin embargo, también podría ser útil buscar aquellos pares cuyo coeficiente de Jaccard sea menor al umbral, es decir, buscar a los pares menos similares.

En la deducción del criterio se llegó a la implicancia de (4.5). Sin embargo también es posible plantear que $\hat{J}_p(A,B) - C_{p,\alpha,n} \leq J(A,B)$ y, por lo tanto,

$$\hat{J}_p(A,B) - C_{p,\alpha,n} > h \implies J(A,B) > h. \tag{5.1}$$

Así, para este problema de encontrar los pares poco similares se puede definir un criterio análogo utilizando la misma cota $C_{p,\alpha,n}$:

Definición 5.1. *Sean $p, n \in \mathbb{N}$ y sean $S_1, S_2 \in \Sigma^*$ dos secuencias genómicas. Entonces, tomando $A = spec_k(S_1)$ y $B = spec_k(S_2)$, y siguiendo la notación de la sub-sección anterior:*

- *Si $\hat{J}_p(A,B) - C_{p,\alpha,n} > h$, entonces el par no se selecciona (se descarta).*

- *Si* $\hat{J}_p(A, B) - C_{p,\alpha,n} \leq h$*, entonces el par sí se selecciona (se conserva).*

Sin embargo, en un conjunto de secuencias se espera que sean pocas las que son parecidas en comparación a las que son poco similares, por lo que el filtrado es mucho más desafiante.

Se usó el conjunto *prefs* para realizar el experimento con distintos tamaños p de los sketches auxiliares, considerando $\alpha = 0.95$ y $n = 1$ para calcular la cota. Las métricas resultantes se muestran en la Cuadro 5.12.

Criterio	ACC	TPR	TNR
hll_4	0.930	1.000	0.102
hll_5	0.933	1.000	0.139
hll_6	0.960	1.000	0.485
hll_7	0.962	1.000	0.511
hll_8	0.990	1.000	0.869
hll_9	0.997	1.000	0.956
hll_10	0.997	1.000	0.968
hll_11	0.998	1.000	0.978
hll_12	0.999	1.000	0.992
hll_13	0.999	1.000	0.991

Cuadro 5.12: Métricas al aplicar el criterio para resolver el problema invertido con el conjunto *prefs*, usando distintos valores de p. Se utilizó $\alpha = 0.95$, $h = 0.2$ y $n = 1$.

Los resultados son bastantes similares a los del problema original, sin embargo se ve que el TNR parte mucho más bajo y recién para $p = 9$ alcanza un valor superior a 0.9. Tal como se esperaba ya que la proporción de pares cuyo Jaccard es cercano a $h = 0.2$ es mucho mayor que la de pares con Jaccard cercano a $h = 0.8$.

Así, el criterio propuesto también puede ser adaptado para enfrentar este problema de filtrado por la izquierda (seleccionar pares con Jaccard bajo un umbral). Sin embargo, hay que tener en cuenta que se requerirá un valor de p mayor para alcanzar precisiones altas sobre los pares negativos.

Capítulo 6

Discusión final

6.1. Conclusiones

La condición de seleccionar los pares de secuencias que cumplan con superar un cierto umbral para su coeficiente de Jaccard, permite crear criterios de selección de modo de no tener que realizar todas las comparaciones posibles. Esto es relevante pues el número de comparaciones crece cuadráticamente con la cantidad de secuencias, y gracias a las nuevas tecnologías, esta crece rápidamente.

El criterio propuesto se muestra efectivo para descartar pares que no superen el umbral, alcanzando una precisión promedio del 98 % sobre los pares negativos (TNR), permitiendo reducir considerablemente el número de comparaciones necesarias. Además, si bien el uso del criterio requiere crear sketches adicionales, se observó que el costo-beneficio de usarlo sigue siendo positivo, según se muestra en las gráficas de aceleración, alcanzando a seleccionar los pares cerca de 9 veces más rápido que si no se ocupara ningún criterio, y cerca de 3 veces más rápido en comparación al criterio directo.

Se cumple así el primer objetivo de esta memoria sobre proponer un criterio basado en Hyperloglog respaldado por el respectivo análisis de error. Además, a través de la implementación y comparación de este nuevo criterio con el criterio directo, se

logran también el segundo y tercer objetivo.

Si bien el contexto de la memoria es similitud genómica, el criterio utilizado realmente sólo pide que los datos sean multiconjuntos. Así, cualquier problema relacionado con similitud de multiconjuntos podría verse beneficiado del nuevo criterio hll_p. Por ejemplo, similitud de textos.

6.2. Trabajo Futuro

Hyperloglog es uno de los tantos sketches que permite trabajar con cardinalidad y similitud. Mash, por ejemplo, utiliza el sketch de MinHash. Así, queda el trabajo de explorar nuevos criterios con diferentes sketches con el fin de agilizar el procesamiento de secuencias y de su similitud. Por otro lado, también es posible explotar la forma en que se realizan las estimaciones para construir nuevos criterios. En particular, la media harmónica que utiliza el sketch de Hyperloglog tiene propiedades que podrían ser usadas para crear un filtro adicional sin necesidad de construir el sketch de mayor tamaño de la unión.

Además, la cota $C_{p,\alpha,n}$ se construye en base a desigualdades que se cumplen con una probabilidad de α. Sin embargo, pareciera que $\alpha = 0.683$ igual da buenos resultados. Esto podría indicar que la cota propuesta se pueda ajustar aún más. Explorar es posibilidad también queda como trabajo futuro.

Bibliografía

[1] Daniel N Baker y Ben Langmead. «Dashing: fast and accurate genomic distances with HyperLogLog». En: *Genome biology* 20.1 (2019), págs. 1-12.

[2] Jessica K Bonnie, Omar Ahmed y Ben Langmead. «DandD: efficient measurement of sequence growth and similarity». En: *bioRxiv* (2023), págs. 2023-02.

[3] Andrei Z Broder. «On the resemblance and containment of documents». En: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE. 1997, págs. 21-29.

[4] Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2022.

[5] Philippe Flajolet y G Nigel Martin. «Probabilistic counting algorithms for data base applications». En: *Journal of computer and system sciences* 31.2 (1985), págs. 182-209.

[6] Philippe Flajolet et al. «Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm». En: *Discrete mathematics & theoretical computer science* Proceedings (2007).

[7] Rob Goldbach. «Genome similarities between plant and animal RNA viruses.» En: *Microbiological sciences* 4.7 (1987), págs. 197-202.

[8] Johan Goris et al. «DNA–DNA hybridization values and their relationship to whole-genome sequence similarities». En: *International journal of systematic and evolutionary microbiology* 57.1 (2007), págs. 81-91.

[9] Stefan Heule, Marc Nunkesser y Alexander Hall. «Hyperloglog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm». En: *Proceedings of the 16th International Conference on Extending Database Technology*. 2013, págs. 683-692.

[10] Chirag Jain et al. «High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries». En: *Nature communications* 9.1 (2018), págs. 1-8.

[11] Tomasz Kociumaka, Gonzalo Navarro y Nicola Prezza. «Towards a definitive measure of repetitiveness». En: *Latin American Symposium on Theoretical Informatics*. Springer. 2020, págs. 207-219.

[12] Sven Kosub. «A note on the triangle inequality for the Jaccard distance». En: *Pattern Recognition Letters* 120 (2019), págs. 36-38.

[13] Camille Marchet et al. «Data structures based on k-mers for querying large collections of sequencing data sets». En: *Genome Research* 31.1 (2021), págs. 1-12.

[14] Azade Nazi et al. «Efficient estimation of inclusion coefficient using hyperloglog sketches». En: *Proceedings of the VLDB Endowment* 11.10 (2018), págs. 1097-1109.

[15] Brian D Ondov et al. «Mash: fast genome and metagenome distance estimation using MinHash». En: *Genome biology* 17.1 (2016), págs. 1-14.

[16] Raimundo Real y J. Vargas. «The Probabilistic Basis of Jaccard's Index of Similarity». En: *Systematic Biology - SYST BIOL* 45 (sep. de 1996), págs. 380-385. DOI: 10.1093/sysbio/45.3.380.

[17] Javier E Soto, Cecilia Hernández y Miguel Figueroa. «JACC-FPGA: A hardware accelerator for Jaccard similarity estimation using FPGAs in the cloud». En: *Future Generation Computer Systems* 138 (2023), págs. 26-42.

[18] Zachary D Stephens et al. «Big data: astronomical or genomical?» En: *PLoS biology* 13.7 (2015), e1002195.

[19] Seok-Hwan Yoon et al. «A large-scale evaluation of algorithms to calculate average nucleotide identity». En: *Antonie Van Leeuwenhoek* 110 (2017), págs. 1281-1286.