Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Repeticiones Maximales para la Estimación de la Diversidad en Metagenomas

Tesis de Licenciatura en Ciencias de la Computación

Maia Tanenzapf

Directores: Esteban Lanzarotti y Pablo Turjanski

Buenos Aires, 2020

REPETICIONES MAXIMALES PARA LA ESTIMACIÓN DE LA DIVERSIDAD EN METAGENOMAS

Los avances en las tecnologías de secuenciación de ADN producidos en la última década permitieron generar grandes cantidades de datos nuevos para analizar. A partir de esto, aplicaciones como la metagenómica tomaron relevancia. Esta consiste en analizar el ADN de los diferentes microorganismos que componen una comunidad a partir de muestras provenientes de distintas fuentes, por ejemplo, de estudios ambientales de suelo o de análisis clínicos de sangre. Las limitaciones actuales de las tecnologías de secuenciación no permiten obtener los genomas completos de los microorganismos presentes en estas muestras, es por esto que uno de los desafíos que se presenta en la actualidad es poder determinar las especies que componen un metagenoma a partir de sus lecturas.

En este trabajo buscamos un método para la estimación de la diversidad de metagenomas de bacterias a partir del cálculo de intervalos maximales de repetición. Para esto utilizamos una adaptación del algoritmo propuesto por Ilie et al. para el cálculo de estos intervalos y analizamos la relación entre distintas propiedades de los mismos y la cantidad de genomas utilizando metagenomas simulados. A partir de este análisis formulamos un modelo que, utilizando los intervalos de repeticiones maximales de un metagenoma, permite estimar la cantidad de genomas que lo integran.

Evaluamos nuestro método en metagenomas simulados a partir de genomas de bacterias conocidos obteniendo una buena estimación de la cantidad de genomas que lo componen. Adicionalmente utilizamos un conjunto de metagenomas de virus generado en el trabajo de Roux et al. de manera de proveer una validacion independiente de los datos usados para obtener el modelo. En este caso obtuvimos un error de escala muy alto al estimar la diversidad, pudiendo deberse a que los datos utilizados fueron generados a partir de virus mientras que nuestro modelo utilizó bacterias o a que en el trabajo de Roux se utilizan entre 500 y 1000 virus mientras que nosotros utilizamos solamente 40 bacterias. Sin embargo, si encontramos cierta correlación al utilizar nuestro método para ordenar los metagenomas según la cantidad de genomas.

Palabras claves: metagenomas, diversidad, repeticiones maximales.

Índice general

1. INTRODUCCIÓN

Los avances en las tecnologías de secuenciación de ADN producidos en la última década cambiaron la manera en la que se producen los datos biológicos. Los costos y los tiempos de secuenciación se redujeron considerablemente permitiendo generar grandes cantidades de datos nuevos para analizar. Estos cambios se produjeron a tal velocidad que la capacidad de secuenciación superó a la ley de Moore. El desarrollo de nuevos métodos que permitan procesar estos datos se volvió necesario, ya que la velocidad en la que aumentan la cantidad y el tamaño de los mismos hace que los algoritmos existentes queden desactualizados rápidamente. Esto supone un reto para las tecnologías de la información [1, 2, 3].

Una de las aplicaciones que tomó relevancia a partir de estos avances es la metagenómica, que consiste en analizar el ADN de los diferentes microorganismos que componen una comunidad a partir de una muestra. Estas secuenciaciones, llamadas metagenomas, pueden provenir de distintas fuentes, por ejemplo, de estudios ambientales de suelo o de análisis clínicos de sangre. Las limitaciones actuales de las tecnologías de secuenciación no permiten obtener los genomas completos de los microorganismos presentes en estas muestras. Sólo es posible obtener metagenomas con lecturas cortas de entre 100 y 500 pares de bases de ADN. Es decir, solo podemos obtener pequeñas partes de cada uno de los genomas presentes. Es por esto que uno de los desafíos que se presenta en la actualidad es poder analizar la diversidad de especies que componen un metagenoma a partir de sus lecturas [4, 5, 6].

La diversidad de un metagenoma se define como la cantidad de especies distintas que lo componen. Descubrimientos recientes muestran que existe una relación entre la diversidad y ciertas enfermedades. Por ejemplo, una baja diversidad se relaciona con el aumento en la frecuencia de muerte en trasplantes alogénicos de células madre [7], representa un marcador biológico para la artritis psoriásica [8] y se asocia con la periodontitis [9]. Por otra parte, una alta diversidad es asociada con infecciones por virus del papiloma humano [10] y con la enfermedad de la banda blanca en los corales [11]. Estos ejemplos muestran que caracterizar a los microorganismos presentes en un metagenoma es un problema fundamental para la biología. Los métodos existentes para la estimación de la diversidad se basan principalmente en métodos paramétricos y no paramétricos. Los métodos paramétricos buscan un modelo sobre ciertas características observadas en muestras previas, por ejemplo la abundancia de ciertas secuencias predeterminadas, que intenta predecir la diversidad en la muestra a determinar. Los no paramétricos requieren del uso de bases de datos de referencia que contengan a todas las especies de la muestra, pero con excepción de algunos casos como el de los humanos, hoy en día esto es difícil de conseguir.

A los fines computacionales, cada lectura de ADN puede ser interpretada como una cadena con los caracteres A, T, C y G. A partir del descubrimiento del genoma humano, el cómputo de repeticiones en una cadena larga de caracteres se volvió un problema importante a ser resuelto. La introducción de una estructura de datos conocida como Suffix Tree impulsó el desarrollo de nuevos algoritmos usados para encontrar distintos tipos de repeticiones de interés para la biología. En 1990, Manber y Myers introdujeron una nueva estructura denominada Suffix Array, que implicaba mejoras en tiempo y espacio en comparación con los algoritmos que usan el Suffix Tree [12, 13].

En nuestro trabajo se utilizará un algoritmo propuesto por Ilie et al. [12] que permite

obtener intervalos maximales de repetición de una cadena de caracteres. A estos intervalos se los denomina repeticiones maximales. El algoritmo de Ilie hace uso de las estructuras de datos Suffix Array y Longest Common Prefix Array para poder encontrar estas repeticiones en tiempo lineal. Un metagenoma se puede representar como un conjunto de cadenas de caracteres donde cada una es una lectura de alguno de los genomas que lo componen. Por lo tanto, extenderemos la definición de repeticiones maximales de Ilie a un conjunto de cadenas de caracteres y utilizaremos una adaptación del algoritmo mencionado anteriormente para poder aplicarlo a todas las lecturas. A partir de este resultado, elegiremos distintas propiedades de estas repeticiones y evaluaremos si varían y de qué manera lo hacen, a medida que cambia la cantidad de genomas presentes. Finalmente, elegiremos una de estas propiedades y a partir de ella intentaremos obtener una función que nos permita estimar la diversidad de un metagenoma.

Para poder hallar una función que arroje resultados satisfactorios, necesitaremos evaluar la correlación entre las propiedades de las repeticiones maximales halladas y la cantidad de genomas presentes en el metagenoma. Para ello, necesitaremos contar con metagenomas de los cuales conozcamos previamente su diversidad. Trabajaremos con metagenomas artificiales generados a partir de una cantidad conocida de genomas. Lo haremos a partir de genomas conocidos de bases de datos públicas y generando los metagenomas tomando subsecuencias al azar de una cantidad preestablecida de dichos genomas. De esta forma podremos variar la cantidad de genomas, la longitud de las lecturas y el tamaño de archivo de los metagenomas generados, para luego analizar la relación entre estas variables y las propiedades de sus repeticiones.

En resumen, intentaremos formular un modelo paramétrico predictivo que, basado en las repeticiones maximales de un metagenoma, permita estimar con precisión la cantidad de genomas que lo integran. Simularemos secuenciaciones construyendo metagenomas artificiales a partir de tomar al azar subsecuencias de genomas de bacterias conocidas. Tomaremos conjuntos de datos de diferentes tamaños, diferentes largos de lecturas y diferentes cantidades de genomas bacterianos, a modo de prueba de concepto, para analizar las propiedades de las repeticiones maximales sobre dichas variables. Evaluaremos la correlación entre las propiedades de dichas repeticiones y la diversidad de genomas presentes en una secuenciación. Finalmente, intentaremos seleccionar una de las propiedades a partir de la cual obtendremos una función que nos permitirá estimar la diversidad de un metagenoma.

2. REPETICIONES MAXIMALES

En este trabajo nos propusimos implementar una heurística paramétrica para estimar la diversidad de un metagenoma basada en el cálculo de repeticiones maximales. Para encontrar las repeticiones maximales utilizaremos una modificación de un algoritmo propuesto por Ilie et al [12]. El algoritmo de Ilie permite obtener todas las repeticiones maximales de una única cadena de caracteres en tiempo lineal con respecto al tamaño de la misma. Utilizaremos una adaptación de este algoritmo que se puede encontrar en la tesis de Rago [17] que permite trabajar con múltiples cadenas de caracteres.

2.1. Definiciones

A continuación presentaremos una serie de definiciones sobre cadenas de caracteres y sobre estructuras de datos útiles para el cálculo de repeticiones.

2.1.1. Cadenas de caracteres

Sea w una cadena de caracteres de longitud n presentamos las siguientes definiciones:

- **Subcadena**
  Llamamos $w[i]$ al caracter en la posición i de w para $1 \leq i \leq n$. Llamamos subcadena $w[i..j]$ a la cadena $w[i]w[i+1]...w[j]$ para $1 \leq i \leq j \leq n$.

- **Prefijo**
  Llamamos $pref_w[i]$ a la cadena $w[1..i]$ para $1 \leq i \leq n$.

- **Sufijo**
  Llamamos $suf_w[i]$ a la cadena $w[i..n]$ para $1 \leq i \leq n$.

Por ejemplo, sea $w = abaababa$, $w[1] = a$, $w[2] = b$, $w[1..1] = a$, $w[1..3] = aba$ y $w[1..8] = abaababa = w$. $pref_w[1] = a$, $pref_w[2] = ab$, $pref_w[5] = abaab$ y $pref_w[n] = w$. $suf_w[1] = abaababa$, $suf_w[2] = baababa$, $suf_w[5] = baba$ y $suf_w[n] = a$.

2.1.2. Repeticiones

- **Repetición**
  Llamamos repetición a una subcadena $w[i..j]$ que aparece dos o más veces en w.

Por ejemplo, para la cadena $w = abaababa$, la subcadena $w[1..1] = a$ es una repetición ya que se repite en $w[1]$, $w[3]$, $w[4]$, $w[6]$ y $w[8]$. La subcadena $w[1..2] = ab$ es una repetición porque $w[1..2] = w[4..5] = w[6..7]$. La subcadena $w[2..4] = baa$ no es una repetición ya que tiene una única aparición.

- **Repetición maximal**
  Llamamos repetición maximal a un **intervalo** $[i..j]$ tal que la subcadena $w[i..j]$ es una repetición y tal que $w[i-1..j]$ y $w[i..j+1]$, si existen, no son repeticiones en w.

Sea $w = abaababa$, el intervalo [4..6] es una repetición maximal en w ya que $w[4..6]$ es una repetición y tanto $w[3..6] = aaba$ como $w[4..7] = abab$ no son repeticiones en w. El intervalo [6..8] también es una repetición maximal porque $w[6..8]$ es una repetición, $w[5..8]$ no es una repetición y $w[6..9]$ no está definido. La cadena $w[1..1]$ es una repetición pero el intervalo [1..1] no es una repetición maximal porque $w[1..2]$ es una repetición.

Notar que, para $w = abcabdabc$, si bien $w[1..2] = w[4..5] = w[7..8] = ab$, el intervalo [4..5] es una repetición maximal pero [1..2] y [7..8] no lo son.

- **Patrón de repetición maximal**
  Llamamos patrón de repetición maximal a una subcadena $w[i..j]$ tal que $[i..j]$ es una repetición maximal en w.

Por ejemplo, para $w = abaababa$ existen 3 repeticiones maximales que son [1..3], [4..6] y [6..8] y un único patrón de repetición maximal que es aba. Las repeticiones maximales de $w = abcabdabc$ son [1..3], [4..5] y [7..9] y sus patrones de repetición maximal son abc y ab.

2.1.3. Repeticiones en conjuntos de cadenas de caracteres

Sea $w_1, ..., w_m$ un conjunto de cadenas de caracteres definimos:

- **Repetición de un conjunto de cadenas**
  Llamamos repetición del conjunto de cadenas a una subcadena $w_k[i..j]$ para $1 \leq k \leq m$ que aparece dos o más veces en las cadenas del conjunto.

Por ejemplo, sean $w_1 = abc$, $w_2 = abc$ y $w_3 = abdd$, $w_1[1..1]$ es una repetición porque $w_1[1..1] = w_2[1..1] = w_3[1..1] = a$. $w_1[3..3]$ es una repetición porque $w_1[3..3] = w_2[3..3] = c$. $w_3[3..3]$ también es una repetición ya que $w_3[3..3] = w_3[4..4]$.

- **Repetición maximal de un conjunto de cadenas**
  Sea $[i..j]_k$ con $1 \leq i \leq j \leq |w_k|$ un intervalo en la cadena k para $1 \leq k \leq m$, llamamos repetición maximal del conjunto de cadenas a un **intervalo** $[i..j]_k$ tal que $w_k[i..j]$ es una repetición en el conjunto de cadenas y tal que $w_k[i-1..j]$ y $w_k[i..j+1]$, si existen, no son repeticiones en el conjunto de cadenas.

Por ejemplo, sean $w_1 = abc$, $w_2 = abc$ y $w_3 = abdd$, el intervalo $[1..3]_1$ es una repetición maximal ya que $w_1[1..3]$ es una repetición y $w_1[0..3]$ y $w_1[1..4]$ no están definidos. $[1..2]_3$ es una repetición maximal porque $w_3[1..2]$ es una repetición, $w_3[0..2]$ no está definido y $w_3[1..3]$ no es una repetición. $[3..3]_3$ es una repetición maximal porque $w_3[3..3]$ es una repetición y $w_3[2..3]$ y $w_3[3..4]$ no lo son.

2.1.4. Suffix array y longest common prefix array

El suffix array (arreglo de sufijos) SA es una estructura de datos que fue introducida por primera vez en 1990 por Manber y Myers [13]. Fue presentada como una alternativa a los suffix trees (árboles de sufijos), que hasta ese momento, eran la estructura más utilizada para búsquedas on line de cadenas de caracteres en textos. La principal ventaja de los suffix arrays sobre los suffix trees fue que ocupaban entre 3 y 5 veces menos espacio. Esta mejora es muy útil en el caso de aplicaciones que trabajan con grandes volúmenes de datos. Sin

embargo, la construcción de los suffix arrays tenía una complejidad de $O(N \log N)$ mientras que la de los suffix trees era de $O(N)$. A lo largo del tiempo, nuevos algoritmos para el suffix array fueron presentados reduciendo tanto la complejidad espacial como la de construcción a lineales. Ejemplos de ello se pueden ver en Ko et al. [14], Kärkkäinen et al. [15] y Nong et al. [16].

El longest common prefix array (arreglo de prefijos comunes más largos) LCP es una estructura auxiliar que se usa para la contrucción SA y se pueden obtener simultáneamente sin modificar la complejidad temporal.

Suffix array

Dada una cadena de caracteres w, el SA de w contiene a las posiciones de todos los sufijos de w ordenados lexicográficamente.

Estos son todos los sufijos de $w = abaababa$:

$$\begin{aligned} suf_w[1] &= abaababa \\ suf_w[2] &= baababa \\ suf_w[3] &= aababa \\ suf_w[4] &= ababa \\ suf_w[5] &= baba \\ suf_w[6] &= aba \\ suf_w[7] &= ba \\ suf_w[8] &= a \end{aligned}$$

Si los ordenamos lexicográficamente y tomamos sus posiciones obtenemos el SA de w:

$$\begin{aligned} suf_w[8] &= a & SA[1] &= 8 \\ suf_w[3] &= aababa & SA[2] &= 3 \\ suf_w[6] &= aba & SA[3] &= 6 \\ suf_w[1] &= abaababa & SA[4] &= 1 \\ suf_w[4] &= ababa & SA[5] &= 4 \\ suf_w[7] &= ba & SA[6] &= 7 \\ suf_w[2] &= baababa & SA[7] &= 2 \\ suf_w[5] &= baba & SA[8] &= 5 \end{aligned}$$

Longest common prefix array

El LCP array es una estructura adicional al SA que contiene en la posición i a la longitud del prefijo común más largo entre $suf_w[SA[i]]$ y $suf_w[SA[i-1]]$. $LCP[i]$ no está definido para $i = 1$.

En la siguiente tabla podemos ver los valores de las posiciones del SA y del LCP de $w = abaababa$:

i	$SA[i]$	$suf_w[SA[i]]$	$LCP[i]$
1	8	a	-
2	3	$\underline{a}ababa$	1
3	6	$\underline{a}ba$	1
4	1	$\underline{aba}ababa$	3
5	4	$\underline{aba}ba$	3
6	7	ba	0
7	2	$\underline{ba}ababa$	2
8	5	$\underline{ba}ba$	2

2.2. Algoritmo

El algoritmo de Ilie et al. [12] permite buscar todas las repeticiones maximales en una única cadena de caracteres. Utilizaremos una extensión del algoritmo formulada en la tesis de Rago [17] para poder trabajar con múltiples cadenas de caracteres. En las siguiente subsecciones daremos una explicación más detallada sobre el funcionamiento e implementación de ambos algoritmos.

2.2.1. Propiedad de prefijos de sufijos

El algoritmo de Ilie et al. [12] se basa en la siguiente propiedad:

Para cualquier repetición maximal $[i..j]$, $w[i..j]$ es el más largo de todos los prefijos repetidos que terminan en j de todos los sufijos de w.

Es decir, si para cada sufijo de w tomamos el prefijo más largo tal que sea una repetición y de esos prefijos tomamos solo los que terminan en una cierta posición, el intervalo asociado al más largo de ellos será una repetición maximal.

Veamos que se cumplen las tres condiciones necesarias para que estos intervalos sean repeticiones maximales. Supongamos que queremos saber si existe una repetición maximal que termina en la posición j. Como tomamos solo prefijos repetidos sabemos que son repeticiones. Como consideramos a los prefijos más largos que son repeticiones, ninguna de esas cadenas podrá extenderse a la derecha y seguir siendo una repetición. Finalmente, como hacemos esto para todos los sufijos vamos a encontrar al prefijo que comienza antes en w, por lo que la cadena no podrá extenderse a la izquierda sin dejar de ser una repetición. Si repetimos esto para todo j desde 1 hasta n obtendremos todas las repeticiones maximales de w.

Veamos un ejemplo para $w = abaababa$. Para cada sufijo, tomamos el prefijo más largo que sea una repetición. Por ejemplo, para $suf_w[1] = abaababa$ podemos ver que $pref_{suf_w[1]}[1] = a$, $pref_{suf_w[1]}[2] = ab$ y $pref_{suf_w[1]}[3] = aba$ son repeticiones en w, pero $pref_{suf_w[1]}[4] = abaa$ no se repite, por lo que el prefijo más largo que es repetición es el que termina en la posición 3. Si hacemos lo mismo para todos los sufijos de w obtenemos los siguientes prefijos:

Sufijo	Prefijo común más largo que es repetición
$suf_w[1] = abaababa$	$w[1..3] = aba$
$suf_w[2] = baababa$	$w[2..3] = ba$
$suf_w[3] = aababa$	$w[3..3] = a$
$suf_w[4] = ababa$	$w[4..6] = aba$
$suf_w[5] = baba$	$w[5..6] = ba$
$suf_w[6] = aba$	$w[6..8] = aba$
$suf_w[7] = ba$	$w[7..8] = ba$
$suf_w[8] = a$	$w[8..8] = a$

Para ver si un intervalo que termina en la posición j es una repetición maximal tenemos que tomar el intervalo del prefijo más largo que termina en esa posición. Para encontrar todas las repeticiones maximales de w repetiremos esto para todo j. Por ejemplo, empezamos por $j = 1$ y vemos que ninguno de los prefijo termina en esa posición. Lo mismo

sucede para $j = 2$. Para la posición $j = 3$, tenemos que tomar el intervalo más largo entre [1..3], [2..3] y [3..3]. El más largo de ellos es [1..3] y por lo tanto es una repetición maximal. Si repetimos esto para todos las valores restantes de j, obtendremos todas las repeticiones maximales de w. Para la posición 6 obtenemos el intervalo [4..6] y para la 8 el [6..8]. Para el resto de las posiciones no hay prefijos para considerar.

2.2.2. Uso de las estructuras auxiliares

El algoritmo propuesto en el trabajo de Ilie et al. [12] se basa en la propiedad de prefijos de sufijos y hace uso de las estructuras SA y LCP para encontrar las repeticiones maximales. El SA de w contiene a las posiciones donde empiezan los sufijos de w ordenados lexicográficamente, por lo que dos elementos contiguos en el SA representan sufijos que pueden tener un prefijo en común. El LCP nos permite saber si existe un prefijo común entre dos elementos contiguos en el SA. Si $LCP[i] = 0$ entonces no hay prefijo común entre $suf_w[SA[i]]$ y $suf_w[SA[i-1]]$. Si $LCP[i] \neq 0$ entonces $suf_w[SA[i]]$ y $suf_w[SA[i-1]]$ tienen un prefijo en común de longitud $LCP[i]$.

El SA y el LCP permiten obtener la longitud del prefijo común más largo entre un sufijo y cualquier otro sufijo de w. Si $LCP[i] \neq 0$ entonces $suf_w[SA[i]]$ y $suf_w[SA[i-1]]$ tienen un prefijo en común. De la misma manera, si $LCP[i+1] \neq 0$ entonces $suf_w[SA[i]]$ y $suf_w[SA[i+1]]$ comparten un prefijo. Si tomamos el mayor entre $LCP[i]$ y $LCP[i+1]$, obtendremos la longitud del prefijo común más largo entre $suf_w[SA[i]]$ y cualquier otro sufijo de w, ya que si existiera otro sufijo con un prefijo común con $suf_w[SA[i]]$ de longitud mayor, entonces el SA no estaría ordenado lexicográficamente. En resumen, si tomamos el máximo entre $LCP[i]$ y $LCP[i+1]$, obtenemos la longitud del prefijo común más largo entre $suf_w[SA[i]]$ y cualquier otro sufijo de w.

El SA y el LCP permiten obtener todas las repeticiones maximales de w. Como vimos anteriormente, podemos obtener el prefijo común más largo entre un sufijo y cualquier otro sufijo de w. Si hacemos esto para todos los sufijos de w podremos evaluar si existe una repetición maximal empezando en cada posición de w. Luego, si para cada repetición encontrada calculamos la posición j donde termina y para cada j nos quedamos con la repetición más larga que termina en esa posición, obtendremos todas las repeticiones maximales de w.

2.2.3. Cómputo de repeticiones maximales

En el *Algoritmo* 2.1 presentamos el algoritmo propuesto por Ilie et al. [12]. Dada una cadena w de longitud n, el resultado del mismo será un arreglo de longitud n llamado $maxRep$ donde:

$$maxRep[j] = \begin{cases} i \text{ si } [i..j] \text{ es una repetición maximal} \\ n+1 \text{ si no hay repetición maximal que termine en la posición } j \end{cases}$$

Primero se computan el SA y el LCP para la cadena w y se inicializan todas las posiciones del arreglo $maxRep$ en $n+1$. Este valor es inválido y se usa para indicar que no hay una repetición maximal que termine en esa posición. El arreglo $maxRep$ se actualizará durante la ejecución del algoritmo para finalmente guardar todas las repeticiones maximales de w como explicaremos a continuación.

Las repeticiones maximales se calculan en el ciclo que comienza en la línea 6. En la

```
1. maxRepeat(w):
2.    compute SA, LCP
3.    n ← length(w)
4.    for k from 1 to n do:
5.        maxRep[k] ← n + 1
6.    for k from 1 to n do:
7.        lcp ← max(LCP[k], LCP[k + 1])
8.    if lcp > 0:
9.        j ← SA[k] + lcp - 1
10.       i ← min(maxRep[j], SA[k])
11.       maxRep[j] ← i
12.   return maxRep
```

Algoritmo 2.1: Algoritmo de Ilie et al. [12] para el cálculo de repeticiones maximales

iteración k se evalúa si existe una repetición maximal tal que su subcadena asociada sea prefijo de $suf_w[SA[k]]$. En la línea 7 se obtiene la longitud lcp del prefijo común más largo entre $suf_w[SA[k]]$ y todos los otros sufijos de w a partir del uso del LCP. Si $lcp > 0$ entonces existe una repetición que comienza en $SA[k]$ y se evalúa si es una repetición maximal. Para eso se calcula la posición j en la que termina la repetición encontrada. Una vez que ya contamos con la posición de inicio y de fin del intervalo se puede evaluar si hay que actualizar $maxRep[j]$. Para eso, se calcula la posición i como el mínimo entre $SA[k]$ y el valor actual de $maxRep[j]$. Si $SA[k]$ es menor que $maxRep[j]$, significa que encontramos una repetición que termina en j y que comienza antes que la encontrada hasta el momento. Finalmente, se asigna i a $maxRep[j]$ que solo cambiará su valor si $SA[k]$ está antes que $maxRep[j]$. Luego de cada iteración se habrán considerado los sufijos hasta la posición k del SA.

Finalmente, luego de evaluar a todos los prefijos, tendremos todas las repeticiones maximales en el arreglo $maxRep$.

2.3. Adaptación del algoritmo para conjuntos de cadenas

2.3.1. Motivación

A los fines computacionales, una lectura de ADN puede ser interpretada como una cadena con los caracteres A, T, C y G. Un metagenoma se puede representar como una lista de cadenas de caracteres donde cada una es una lectura de los genomas que lo componen. En este trabajo buscaremos un modelo que permita estimar la diversidad de un metagenoma a partir de sus repeticiones maximales. Por lo tanto, utilizaremos una adaptación del algoritmo propuesto por Ilie et al. [12] que se puede encontrar en la tesis de Rago [17] para calcular las repeticiones maximales de un conjunto de cadenas. Vamos a utilizar este nuevo algoritmo tomando como entrada el conjunto de lecturas de un metagenoma y de esta forma obtendremos las repeticiones maximales del mismo.

2.3.2. Adaptación del algoritmo

El algoritmo de Ilie et al. encuentra las repeticiones maximales dada una sola cadena de caracteres. Nos interesa considerar muchas cadenas, por lo que utilizaremos una mo-

dificación del algoritmo para poder obtener las repeticiones maximales de un conjunto de cadenas. Esta adaptación consiste en modificar la entrada del algoritmo y la construcción del LCP.

El algoritmo original toma una única cadena de caracteres como entrada. Dado un conjunto de cadenas w_1,...,w_m construímos la cadena w a partir de concatenar todas las cadenas con un caracter especial # que no pertenece al alfabeto. Es decir, la nueva entrada del algoritmo será una única cadena $w = w_1\#w_2\#...\#w_m$.

Como vimos anteriormente, una repetición maximal de w_1,...,w_m es una cadena que está completamente contenida en algunas de las cadenas del conjunto. Es decir, una repetición maximal de w_1,...,w_m no puede contener al caracter #. Vamos a adaptar la construcción del LCP para que al utilizarlo para buscar repeticiones no se consideren aquellas que contienen al caracter especial #. Para esto, basta con considerar todas las apariciones del caracter # como caracteres distintos entre sí durante la construcción del LCP.

Veamos una comparación del LCP original con el LCP si aplicamos la modificación anterior. Sean $w_1 = abc$, $w_2 = adc$, $w_3 = ab$ y $w = abc\#adc\#ab$, el LCP y el LCP modificado de w son:

i	$SA[i]$	$suf_w[SA[i]]$	$LCP[i]$
1	8	$\#ab$	-
2	4	$\#adc\#ab$	2
3	9	ab	0
4	1	$abc\#adc\#ab$	2
5	5	$adc\#ab$	1
6	10	b	0
7	2	$bc\#adc\#ab$	1
8	7	$c\#ab$	0
9	3	$c\#adc\#ab$	3
10	6	$dc\#ab$	0

i	$SA[i]$	$suf_w[SA[i]]$	$LCP[i]$
1	8	$\#ab$	-
2	4	$\#adc\#ab$	**0**
3	9	ab	0
4	1	$abc\#adc\#ab$	2
5	5	$adc\#ab$	1
6	10	b	0
7	2	$bc\#adc\#ab$	1
8	7	$c\#ab$	0
9	3	$c\#adc\#ab$	**1**
10	6	$dc\#ab$	0

Se puede ver que en el segundo LCP no se consideran las cadenas que tienen el caracter #. Por ejemplo, SA[8] y SA[9] tienen un prefijo común de longitud 3 en el primer LCP y uno de longitud 1 en el LCP modificado.

Esta modificación permite que, al consultar el LCP en la línea 7 del algoritmo para buscar una repetición, no se consideren repeticiones que contengan al caracter #.

En resumen, para calcular todas las repeticiones maximales de w_1,...,w_m construímos la cadena w a partir de la concatenación de todas las cadenas usando el caracter especial # y la usamos como entrada del algoritmo modificado.

2.3.3. Orden algorítmico

La adaptación del algoritmo obtiene todas las repeticiones maximales de la cadena $w = w_1\#w_2\#...\#w_m$ con complejidad $O(n)$ siendo n el tamaño de la cadena w.

Primero se construyen el SA y el LCP de esa cadena. Ambos pueden construirse simultáneamente con una complejidad temporal lineal con respecto a n [14] [15] [16].

Luego se inicializa el arreglo $maxRep$ que tiene tamaño n por lo que este paso tiene complejidad $O(n)$.

Finalmente, se realizan n iteraciones donde solo se acceden a las estructuras construídas anteriormente y se hacen comparaciones entre sus valores por lo que todo este ciclo también

tiene complejidad $O(n)$.

Así, el algoritmo para el cómputo de las repeticiones maximales de un conjunto de cadenas tiene una complejidad temporal lineal con respecto a la longitud de w.

En cuanto a la complejidad espacial, el SA, el LCP y el arreglo $maxRep$ tienen complejidad espacial lineal. Por lo tanto, todo el algoritmo tiene complejidad espacial lineal.

3. PROPIEDADES DE LAS REPETICIONES MAXIMALES EN METAGENOMAS SIMULADOS

Dado un metagenoma, queremos estimar la cantidad de genomas distintos que lo componen a partir del cálculo de sus repeticiones maximales. Evaluaremos si existen propiedades de las repeticiones maximales de un metagenoma que se relacionen con la cantidad de genomas del mismo. Para esto, utilizaremos metagenomas artificiales que simularemos tomando al azar subsecuencias de genomas de bacterias conocidos, de manera que nos permita conocer previamente la cantidad de genomas que los componen. Generaremos metagenomas combinando diferentes tamaños, largos de lecturas y cantidades de genomas. Finalmente, elegiremos una de estas propiedades para obtener, a partir de ella, una función que nos permita estimar la diversidad de un metagenoma.

3.1. Generación de metagenomas artificiales

3.1.1. Generación de metagenomas

Un metagenoma contiene una lista de lecturas de los genomas que lo componen. Para evaluar si existe una relación entre una propiedad de las repeticiones maximales y la cantidad de genomas necesitamos contar con metagenomas de los cuales conozcamos su diversidad. Para esto generaremos metagenomas artificiales a partir de un conjunto de genomas obtenidos de bases de datos públicas que se puede ver en la sección $A.1$ del apéndice. A partir de este conjunto, podremos simular un metagenoma tomando subsecuencias al azar de los distintos genomas presentes en el mismo. Como método de control para el análisis de propiedades generaremos metagenomas *scrambled*. Estos serán generados de la misma manera que los metagenomas pero cada una de las lecturas seleccionadas será desordenada, con el objetivo de eliminar las posibles repeticiones presentes en las mismas. En el *Algoritmo* 3.1 vemos un pseudocódigo del script `simul-meta.py` utilizado para la generación de metagenomas:

```
1. generateMetagenome(genomes, readlen, size, scrambled):
2.     artificial_metagenome ← empty metagenome
3.     while size > 0:
4.         random_genome ← genomes[randint(1, length(genomes))]
5.         random_position ← randint(1, length(random_genome) - readlen + 1)
6.         random_lecture ← random_genome[random_position..random_position +
readlen - 1]
7.         if scrambled:
8.             random_lecture ← scramble(random_lecture)
9.         add(artificial_metagenome, random_lecture)
10.        size ← size - readlen
11.    return artificial_metagenome
```

Algoritmo 3.1: Algoritmo utilizado para la generación de metagenomas simulados

El script anterior toma una lista de genomas `genomes`, una longitud de lectura `readlen`, un tamaño `size` para el archivo a generar y un parámetro `scrambled` mediante el cual permite especificar si el metagenoma generado será de tipo scrambled o no. Además, contamos con las siguientes funciones auxiliares: `randint` que permite elegir un valor aleatorio entre dos valores dados, `length` que calcula la longitud de una cadena de caracteres, `scramble` que desordena los caracteres de una cadena y `add` para agregar una lectura a un metagenoma. Primero se inicializa un metagenoma sin lecturas llamado `artificial_metagenome` al que se le agregarán lecturas con la longitud especificada hasta completar el tamaño de archivo requerido. En cada iteración se elige un genoma `random_genome` de manera equiprobable de la lista de genomas. Luego se selecciona una posición al azar de `random_genome` utilizando la función `randint` a partir de la cual se tomará una lectura `random_lecture` con longitud `readlen`. En caso de que se requiera que el metagenoma generado sea de tipo scrambled, `random_lecture` es desordenada con la función `scramble`. Luego `random_lecture` es agregada a `artificial_metagenome` usando la función `add`. Una vez alcanzado el tamaño de archivo se devuelve el metagenoma `artificial_metagenome`.

3.1.2. Relación entre repeticiones maximales y cantidad de genomas de un metagenoma

Queremos analizar si existe una relación entre las repeticiones maximales y la cantidad de los genomas de un metagenoma. Para eso utilizamos el script `simul-meta.py` para generar metagenomas variando la cantidad de genomas utilizados y fijando la longitud de las lecturas en 100 nucleótidos y el tamaño de archivo en 10 Mb, y calculamos sus repeticiones maximales.

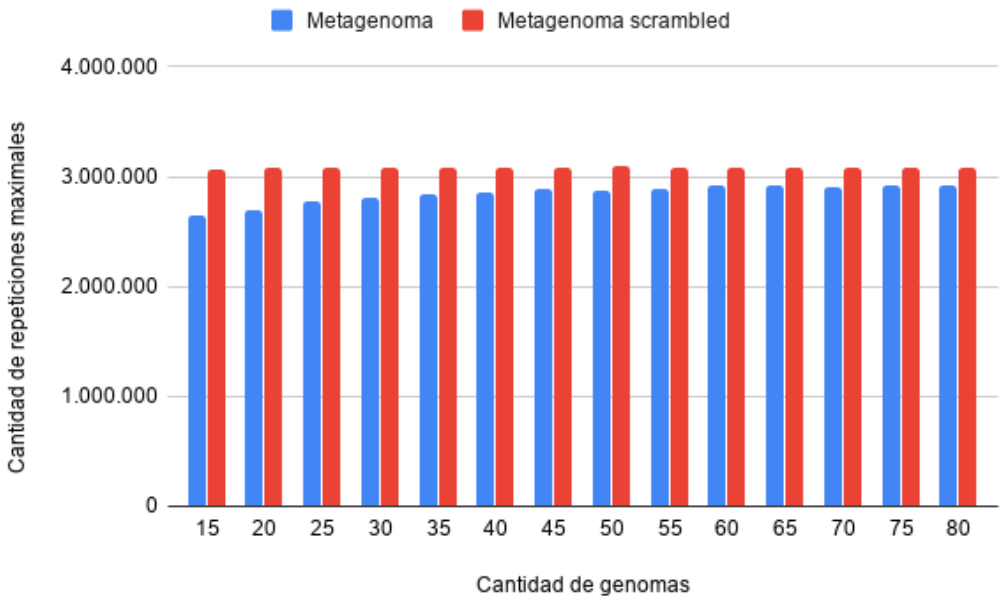

Figura 3.1: Cantidad de repeticiones maximales de un metagenoma generado con lecturas de longitud 100 y tamaño 10 Mb en función de la cantidad de genomas.

En la *Figura* 3.1 vemos en azul los resultados para los metagenomas y en rojo para los

metagenomas scrambled. Podemos ver que en el caso de los metagenomas la cantidad de repeticiones maximales aumenta a medida que aumenta la cantidad de genomas, mientras que se mantiene constante para los metagenomas scrambled. Por lo tanto, podemos hipotetizar que existe una relación entre la cantidad de repeticiones maximales y la cantidad de genomas presentes en un metagenoma.

Para corroborar lo observado en la *Figura* 3.1 utilizaremos los coeficientes de correlación de Spearman y Pearson. Estos coeficientes permiten medir la correlación entre dos variables, en nuestro caso, la cantidad de genomas y la cantidad de repeticiones maximales. En ambos casos el valor del coeficiente de correlación varía en el intervalo [-1,1]. Si su valor es 0, no existe correlación entre las variables, mientras que valores cercanos a -1 o 1 indican correlaciones negativas o positivas respectivamente.

En la *Tabla* 3.1 y la *Tabla* 3.2 vemos los valores de los coeficientes de Spearman y Pearson para los datos de la *Figura* 3.1.

	valor	p-valor
Metagenoma	0,978	1,55e-9
Metagenoma scrambled	0,1034	0,7249

Tabla 3.1: Valores del coeficiente de correlación de Spearman para la cantidad de repeticiones maximales en función de la cantidad de genomas para metagenoma y metagenoma scrambled generados con longitud de lectura 100 y tamaño de archivo de 10 Mb.

	valor	p-valor
Metagenoma	0,9013	1,07e-5
Metagenoma scrambled	0,2549	0,3789

Tabla 3.2: Valores del coeficiente de correlación de Pearson para la cantidad de repeticiones maximales en función de la cantidad de genomas para metagenoma y metagenoma scrambled generados con longitud de lectura 100 y tamaño de archivo de 10 Mb.

En el caso de los metagenomas, para ambos coeficientes obtuvimos valores muy cercanos a 1 indicando que existe una correlación entre la cantidad de genomas y la cantidad de repeticiones maximales. En cambio, para los metagenomas scrambled, los valores son cercanos a 0 por lo que concluímos que no existe esta correlación. Dado que según nuestros datos existe correlación entre estas variables, intentaremos encontrar una función que permita estimar la diversidad a partir del cálculo de las repeticiones maximales.

3.1.3. Análisis de los parámetros para la generación de metagenomas

El script utilizado para la generación de metagenomas nos permite especificar la longitud de las lecturas y el tamaño de archivo. Realizaremos un estudio exploratorio de cómo varían las repeticiones maximales en relación a distintos parámetros posibles. Para esto generaremos metagenomas fijando la cantidad de genomas en 80 y variaremos los otros dos parámetros recibidos por el script `generateMetagenome`.

Tamaño de archivo

Primero analizaremos qué sucede si variamos el tamaño de archivo. Con el script `simul-meta.py`, descripto anteriormente, generamos metagenomas y metagenomas scram-

bled utilizando 80 genomas, longitud de lectura 100 y variando el tamaño del archivo. Utilizamos archivos de tamaño 5, 10 y 20 Mb y graficamos la cantidad de repeticiones maximales en función del tamaño de archivo.

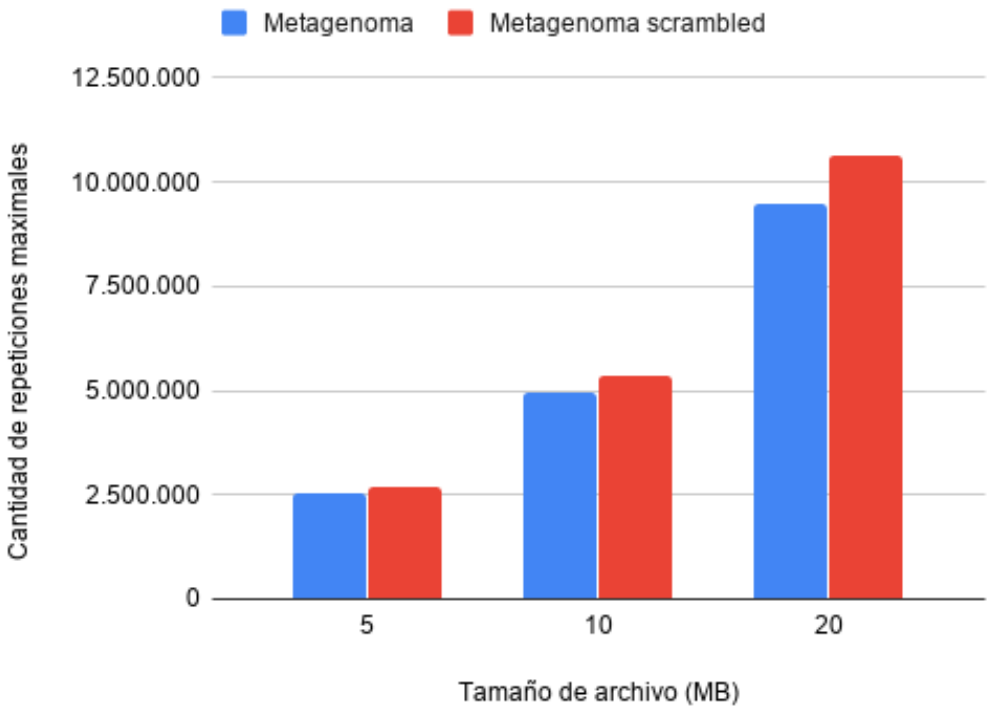

Figura 3.2: Cantidad de repeticiones maximales en función del tamaño de archivo para metagenomas generados a partir de 80 genomas y con lecturas de longitud 100.

Podemos ver en la *Figura* 3.2 que la cantidad de repeticiones maximales es mayor para los metagenomas scrambled. También observamos que la cantidad aumenta si aumentamos el tamaño de archivo y que lo hace en mayor proporción para los metagenomas scrambled.

Longitud de las lecturas

En este caso analizaremos qué pasa si variamos la longitud de las lecturas. Debido a las limitaciones de las tecnologías actuales, los metagenomas están compuestos de lecturas cortas de entre 100 y 500 pares de bases de ADN, por lo que utilizaremos valores dentro de ese rango. Analizaremos qué sucede con la cantidad de repeticiones maximales a medida que utilizamos diferentes longitudes de lecturas.

Generamos metagenomas y metagenomas scrambled utilizando 80 genomas, tamaño de archivo 10 Mb y utilizando lecturas de longitud 100, 200 y 400 y graficamos la cantidad de repeticiones maximales en función de la longitud de las lecturas.

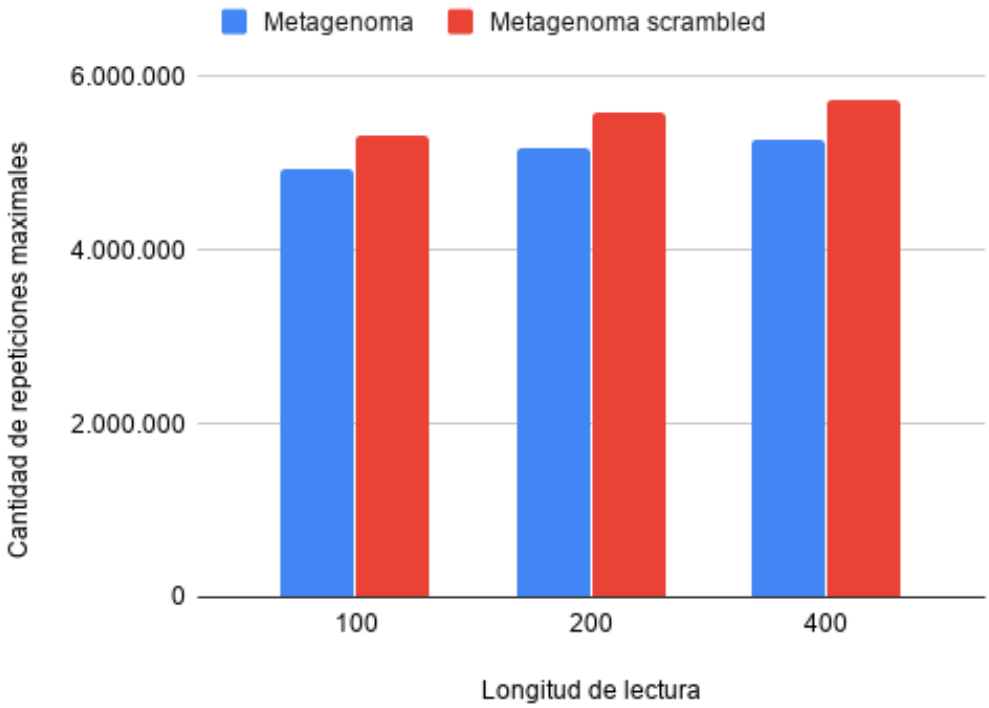

Figura 3.3: Cantidad de repeticiones maximales en función de la longitud de sus lecturas para metagenomas generados a partir de 80 genomas y con tamaño 10 Mb.

En la *Figura* 3.3 vemos que a medida que aumentamos la longitud de las lecturas obtenemos una mayor cantidad de repeticiones maximales para ambos tipos de metagenomas. También encontramos una mayor cantidad de repeticiones maximales en el caso de los metagenomas scrambled.

En ambos casos la cantidad de repeticiones maximales es mayor para los metagenomas scrambled que para los metagenomas. Esto puede deberse a que al mezclar aleatoriamente las lecturas de los metagenomas scrambled estamos generando repeticiones que no estarían presentes en los metagenomas.

3.2. Propiedades

Analizaremos distintas propiedades relacionadas con las repeticiones maximales y evaluaremos su correlación con la cantidad de genomas de un metagenoma. Para esto generaremos conjuntos de metagenomas con distinta cantidad de genomas y para cada uno calcularemos sus repeticiones maximales y luego el valor de las distintas propiedades a analizar. Finalmente graficaremos los resultados de las propiedades en función de la cantidad de genomas y calcularemos las correlaciones de Spearman y Pearson para elegir la propiedad que arroje mejores resultados. Realizaremos el mismo procedimiento para metagenomas scrambled y compararemos los resultados.

3.2.1. Generación de metagenomas para el análisis de propiedades

Para realizar el análisis de las propiedades a evaluar generaremos distintos conjuntos de metagenomas variando la cantidad de genomas, el tamaño de archivo y la longitud de las lecturas. Luego calcularemos el valor para distintas propiedades y las graficaremos en función de la cantidad de genomas. Finalmente calcularemos los coeficientes de Spearman y Pearson. El proceso para obtener los resultados para las distintas propiedades puede verse en la *Figura* 3.4.

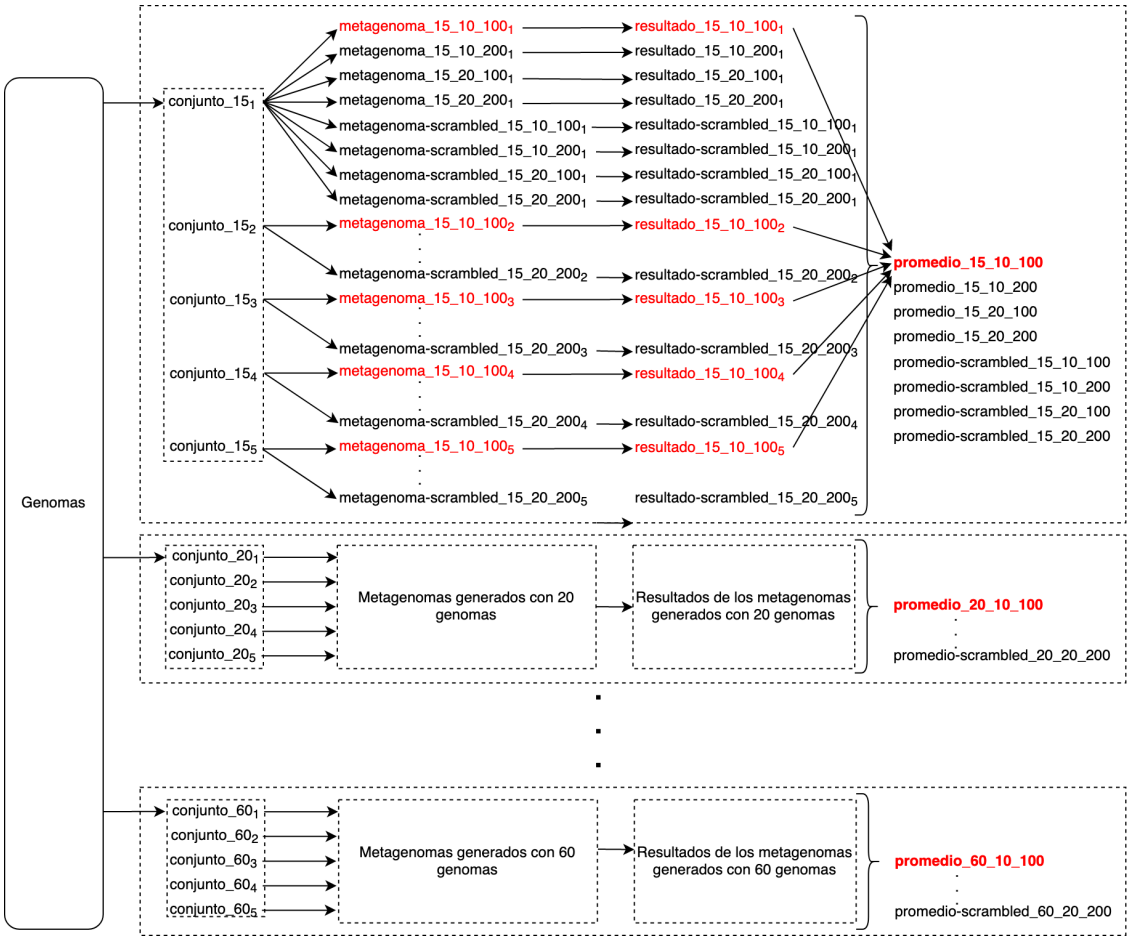

Figura 3.4: Proceso de generación de conjuntos de metagenomas y de resultados para el análisis de propiedades.

Como primer paso tomamos conjuntos de distintas cantidades de genomas a partir de un conjunto de 80 genomas. Para cada una de esas cantidades armamos 5 conjuntos distintos. Estos pueden verse en el gráfico representados como $conjunto\_x_i$ donde $x \in \{15, 20, 25, 30, 35, 40, 45, 50, 55, 60\}$ es la cantidad de genomas del conjunto y $1 \leq i \leq 5$ representa el número del mismo.

El siguiente paso es la generación de los metagenomas a partir de los conjuntos seleccionados. Por cada uno generamos 4 metagenomas y 4 metagenomas scrambled combinando 10 Mb y 20 Mb como tamaño de archivo y 100 y 200 como longitud de lectura. Es decir, generamos 4 $metagenoma\_x\_t\_l_i$ y 4 $metagenoma\text{-}scrambled\_x\_t\_l_i$, donde x es la cantidad de genomas, $t \in \{10, 20\}$ es el tamaño de archivo, $l \in \{100, 200\}$ es la longitud

de las lecturas e i es el número del conjunto. Por ejemplo, para el conjunto $conjunto\_15_1$ generamos los siguientes 8 metagenomas:

1. $metagenoma\_15\_10\_100_1$
2. $metagenoma\_15\_10\_200_1$
3. $metagenoma\_15\_20\_100_1$
4. $metagenoma\_15\_20\_200_1$
5. $metagenoma\text{-}scrambled\_15\_10\_100_1$
6. $metagenoma\text{-}scrambled\_15\_10\_200_1$
7. $metagenoma\text{-}scrambled\_15\_20\_100_1$
8. $metagenoma\text{-}scrambled\_15\_20\_200_1$

Una vez obtenidos los metagenomas los utilizaremos para evaluar las distintas propiedades propuestas. Para todos los metagenomas calcularemos el valor de la propiedad a evaluar y obtendremos un $resultado\_x\_t\_l_i$ por cada $metagenoma\_x\_t\_l_i$ y un $resultado\text{-}scrambled\_x\_t\_l_i$ por cada $metagenoma\text{-}scrambled\_x\_t\_l_i$. Por ejemplo, para cada uno de los siguientes metagenomas obtenemos los siguientes resultados:

1. $metagenoma\_15\_10\_100_1 \longrightarrow resultado\_15\_10\_100_1$
2. $metagenoma\_15\_10\_200_1 \longrightarrow resultado\_15\_10\_200_1$
3. $metagenoma\_15\_20\_100_1 \longrightarrow resultado\_15\_20\_100_1$
4. $metagenoma\_15\_20\_200_1 \longrightarrow resultado\_15\_20\_200_1$
5. $metagenoma\text{-}scrambled\_15\_10\_100_1 \longrightarrow resultado\text{-}scrambled\_15\_10\_100_1$
6. $metagenoma\text{-}scrambled\_15\_10\_200_1 \longrightarrow resultado\text{-}scrambled\_15\_10\_200_1$
7. $metagenoma\text{-}scrambled\_15\_20\_100_1 \longrightarrow resultado\text{-}scrambled\_15\_20\_100_1$
8. $metagenoma\text{-}scrambled\_15\_20\_200_1 \longrightarrow resultado\text{-}scrambled\_15\_20\_200_1$

Luego calcularemos el promedio de los valores correspondientes a los metagenomas y metagenomas scrambled generados con misma cantidad de genomas y mismos parámetros. De esta forma obtendremos un $promedio\_x\_t\_l$ y un $promedio\text{-}scrambled\_x\_t\_l$ por cada cantidad de genomas utilizada para generar los conjuntos iniciales. Por ejemplo, para obtener $promedio\_15\_10\_100$ utilizaremos los siguientes resultados:

1. $resultado\_15\_10\_100_1$
2. $resultado\_15\_10\_100_2$
3. $resultado\_15\_10\_100_3$
4. $resultado\_15\_10\_100_4$
5. $resultado\_15\_10\_100_5$

y para obtener $promedio\text{-}scrambled\_15\_10\_100$ utilizaremos:

1. $resultado\text{-}scrambled\_15\_10\_100_1$
2. $resultado\text{-}scrambled\_15\_10\_100_2$
3. $resultado\text{-}scrambled\_15\_10\_100_3$
4. $resultado\text{-}scrambled\_15\_10\_100_4$
5. $resultado\text{-}scrambled\_15\_10\_100_5$

Finalmente graficaremos todos los $promedio\_x\_t\_l$ en función de x y calcularemos los coeficientes de Spearman y Pearson de manera que podamos analizar la correlación entre la cantidad de genomas y las distintas propiedades.

3.2.2. Propiedades

Propiedad 1: Cantidad de repeticiones maximales

En este caso analizaremos la propiedad de la cantidad de repeticiones maximales de un metagenoma.

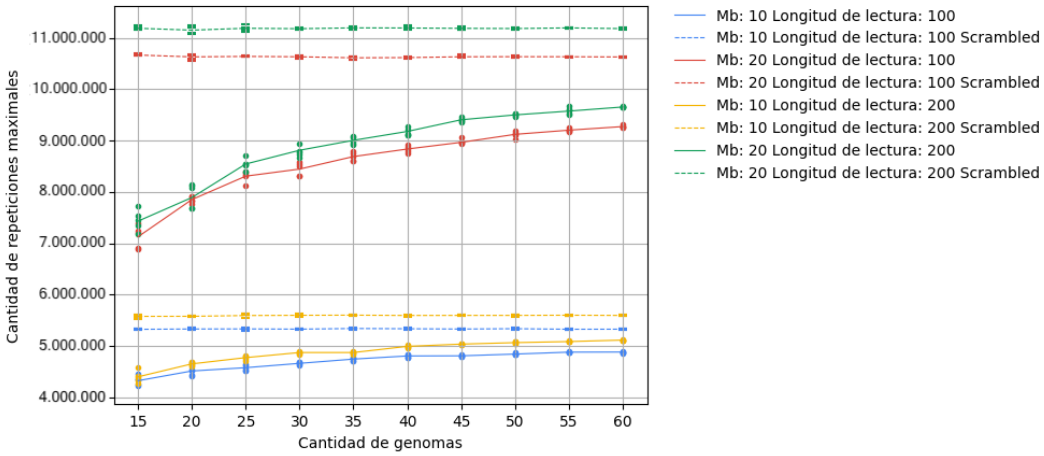

Figura 3.5: Cantidad de repeticiones maximales en función de la cantidad de genomas para conjuntos de metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

En la *Figura* 3.5 podemos ver los resultados de la Propiedad 1 obtenidos para metagenomas y metagenomas scrambled generados con distintos parámetros. Para sintetizar los resultados obtenidos, decidimos superponer los gráficos obtenidos para las diferentes combinaciones de parámetros. Realizaremos el mismo gráfico para todas las propiedades a evaluar de manera que podamos compararlas.

Para cada cantidad de genomas graficamos los 5 resultados obtenidos. Los puntos representan los valores de los $resultado\_x\_t\_l_i$ y los guiones de los $resultado\text{-}scrambled\_x\_t\_l_i$. Podemos ver que para los metagenomas encontramos valores diferentes mientras que para los metagenomas scrambled los valores obtenidos se superponen siendo casi los mismos.

También podemos ver los gráficos de los $promedio\_x\_t\_l$, que muestran que la cantidad de repeticiones maximales aumenta para los metagenomas y se mantiene constante para los metagenomas scrambled. A partir de esto podemos concluir que existe una correlación entre la cantidad de genomas de un metagenoma y esta propiedad. Para verificar lo observado en la *Figura* 3.5 calculamos los coeficientes de Spearman y Pearson. Podemos ver los resultados obtenidos en para el coeficiente de Spearman en la *Tabla* 3.3 y para el de Pearson en *Tabla* 3.4.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	1	6,64e-64	1	6,64e-64	0,9969	3,69e-20	1	6,64e-64
Metagenoma scrambled	0,006	0,9867	-0,2606	0,467	0,5636	0,0897	0,1272	0,726

Tabla 3.3: Valores del coeficiente de correlación de Spearman para la cantidad de repeticiones maximales en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,941	4,91e-05	0,9368	6,44e-05	0,9341	7,6e-05	0,9448	3,77e-05
Metagenoma scrambled	0,0384	0,916	-0,4389	0,2043	0,6973	0,0249	0,3443	0,3298

Tabla 3.4: Valores del coeficiente de correlación de Pearson para la cantidad de repeticiones maximales en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados ccon distintas combinaciones de tamaño de archivo y de longitud de lectura.

En cada columna de la *Tabla* 3.3 y la *Tabla* 3.4 vemos el valor y el p-valor obtenido para una combinación de tamaño de archivo y de longitud de lectura posible. En la primera fila mostramos los valores obtenidos para los metagenomas y en la segunda para los metagenomas scrambled. En ambos casos vemos valores cercanos a 1 para los metagenomas y valores cercanos a 0 para los metagenomas scrambled. De esta forma observamos que existe una correlación con la cantidad de genomas para los metagenomas, mientras que no existe para los metagenomas scrambled.

Propiedad 2: Cantidad de patrones de repeticiones maximales

A continuación analizaremos la propiedad de la cantidad de patrones de repetición maximal. La cantidad de patrones es siempre menor que la cantidad de instancias, por lo que esperamos un gráfico similar al anterior pero con valores menores. En la *Figura* 3.6 podemos ver los resultados de la Propiedad 2.

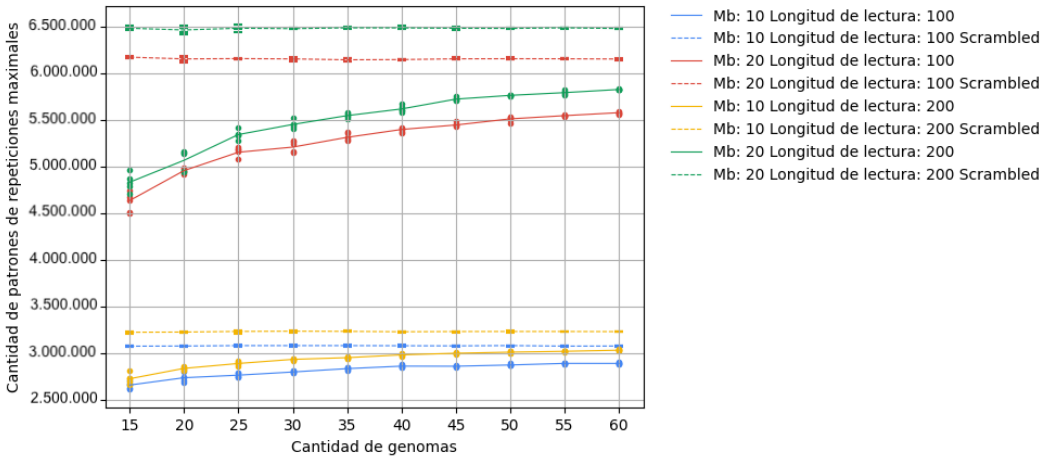

Figura 3.6: Cantidad de patrones de repetición maximal en función de la cantidad de genomas para conjuntos de metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Nuevamente vemos una correlación solamente para los metagenomas. Esto se puede verificar con los valores de los coeficientes de Spearman y Pearson que se muestran en la *Tabla* 3.5 y la *Tabla* 3.6.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,9757	1,46e-06	1	6,64e-64	1	6,64e-64	1	6,64e-64
Metagenoma scrambled	0,0184	0,9597	-0,2606	0,467	0,1757	0,6271	0,1272	0,726

Tabla 3.5: Valores del coeficiente de correlación de Spearman para la cantidad de patrones de repetición maximal en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,9393	5,49e-05	0,9371	6,31e-05	0,9231	0,0001	0,9425	4,45e-05
Metagenoma scrambled	0,0173	0,9621	-0,4203	0,2264	0,4523	0,1893	0,369	0,2939

Tabla 3.6: Valores del coeficiente de correlación de Pearson para la cantidad de patrones de repetición maximal en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Igual que en el caso anterior, observamos en la *Tabla* 3.5 y la *Tabla* 3.6 que para la Propiedad 2 existe una correlación con la cantidad de genomas en los metagenomas y que no existe para los metagenomas scrambled.

Cantidad de repeticiones maximales vs. cantidad de patrones

Hasta ahora analizamos dos propiedades, la cantidad de repeticiones maximales y la cantidad de patrones de repetición maximal. Ambas propiedades arrojaron resultados similares como vimos en los gráficos anteriores y en los valores de los coeficientes de Spearman y Pearson. Sin embargo, el cálculo de la cantidad de patrones es más costoso que el de la cantidad de repeticiones maximales. El algoritmo utilizado calcula las repeticiones maximales y requiere un procesamiento adicional en caso de que se quieran obtener los patrones.

Dado que los resultados son muy similares y que el cálculo de la cantidad de repeticiones maximales no requiere este procesamiento adicional, continuaremos analizando propiedades solamente sobre estas. En particular, analizaremos a continuación propiedades relacionadas a la cantidad de repeticiones maximales en función de sus longitudes.

Cantidad de repeticiones maximales en función de sus longitudes

Al calcular las repeticiones maximales de un metagenoma podemos obtener intervalos de longitud entre 1 y la longitud de sus lecturas. Dada una longitud, podemos contar la cantidad de repeticiones maximales de ese largo. Llamamos $r_i(metagenoma)$ a la cantidad de repeticiones maximales de longitud i del metagenoma. En la *Figura* 3.7 vemos un histograma de los valores de los $r_i(metagenoma)$ para un metagenoma generado con longitud de lectura 100 y tamaño de archivo 10 Mb.

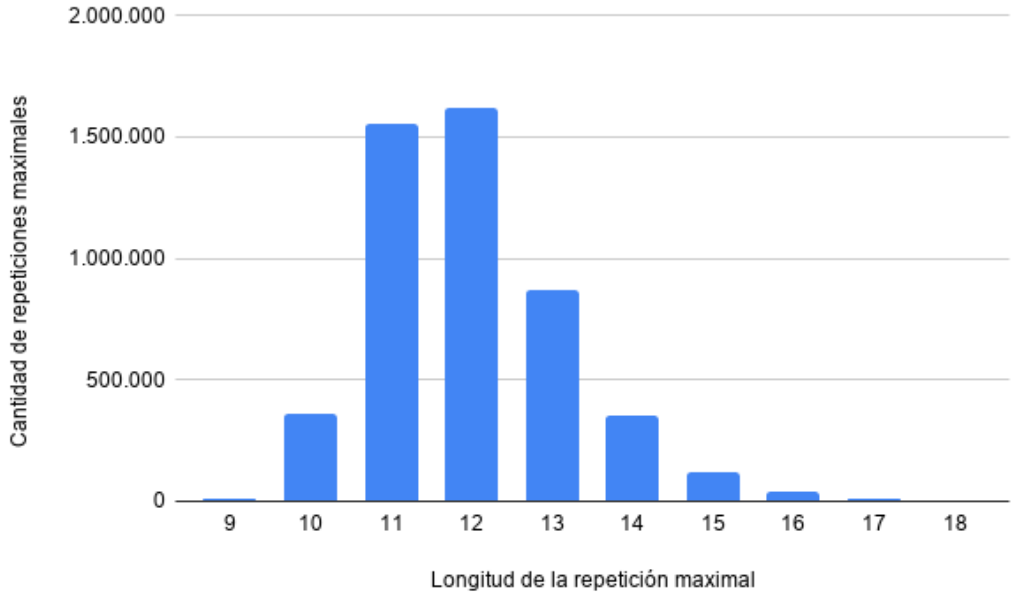

Figura 3.7: Histograma de la longitud de las repeticiones maximales para un metagenoma simulado con longitud de lectura 100 y tamaño de archivo de 10 Mb.

En la *Figura* 3.7 observamos que la mayoría de las repeticiones se distribuyen entre una longitud de 10 y 15 caracteres, teniendo su pico máximo en 12 caracteres.

Realizamos la misma prueba para metagenomas y metagenomas scrambled generados con distinta cantidad de genomas y graficamos la cantidad de repeticiones maximales en función del largo de la repetición. En particular nos centramos en longitudes de repetición en el rango que va de 1 a 20 caracteres dado que tanto en la figura anterior como a lo largo los siguientes experimentos observamos que la cantidad de repeticiones maximales es despreciable para longitudes más allá de estos valores.

En la *Figura* 3.8 vemos cuatro gráficos donde cada uno de ellos agrupa los resultados de distintos metagenomas y metagenomas scrambled generados con la misma longitud de lectura y tamaño de archivo. Graficamos las longitudes de las repeticiones maximales entre 8 y 20 en el eje x y la cantidad de repeticiones maximales que corresponden a ese largo en el eje y. De nuevo observamos que la mayoría de las repeticiones tienen una longitud de entre 10 y 15 caracteres y que el pico se encuentra en 11 o 12 caracteres. También podemos ver, sobre todo en los gráficos correspondientes a los archivos de mayor tamaño representados en la fila inferior, que los gráficos de los metagenomas se solapan menos que los de los metagenomas scrambled. Si bien no se observa en el gráfico, pudimos encontrar que para los metagenomas scrambled las longitudes de las repeticiones maximales encontradas están entre 8 y 20 caracteres, mientras que para los metagenomas encontramos repeticiones de todas las longitudes posibles, incluso de las longitudes máximas.

A continuación presentaremos distintas propiedades relacionadas a la cantidad de repeticiones maximales en función de su longitud.

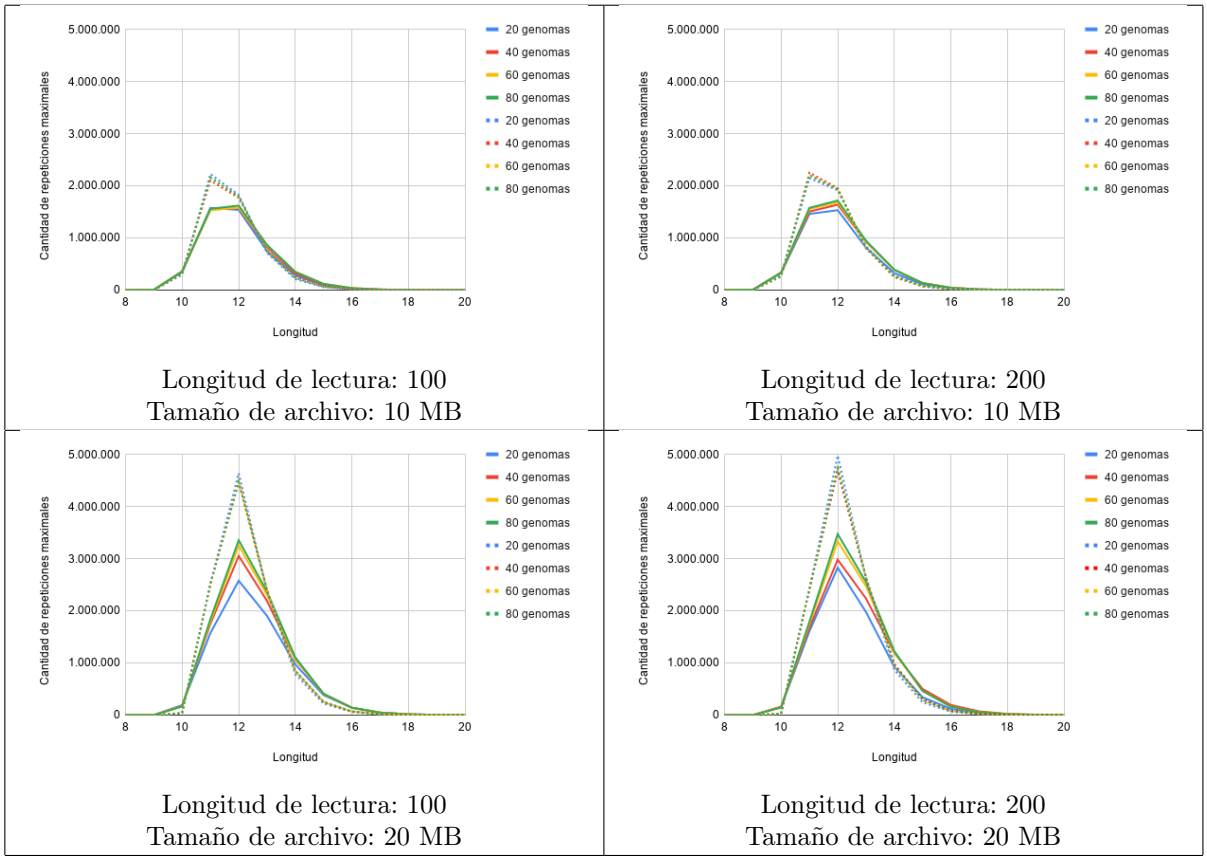

Figura 3.8: Cantidad de repeticiones maximales en función de su longitud para metagenomas y metagenomas scrambled generados con distintas combinaciones de longitud de lectura y tamaño de archivo.

Propiedad 3: Cantidad de repeticiones maximales de longitud hasta 15

Como vimos en la *Figura* 3.7 y la *Figura* 3.8, las repeticiones tienen en su mayoría una longitud de hasta 15 caracteres. Vamos a evaluar la propiedad de la cantidad de repeticiones maximales de longitud menor o igual a 15. Sea $r_i(metagenoma)$ la cantidad de repeticiones maximales de longitud i de un metagenoma, se detalla la fórmula aplicada:

$$cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15(metagenoma) =$$

$$\sum_{i=1}^{15} r_i(metagenoma)$$

Esperamos obtener valores muy similares a los del análisis de la cantidad de instancias, ya que estamos evaluando la cantidad de instancias de longitud menor o igual a 15, y ya vimos que la cantidad de instancias con longitudes mayores a 15 es despreciable.

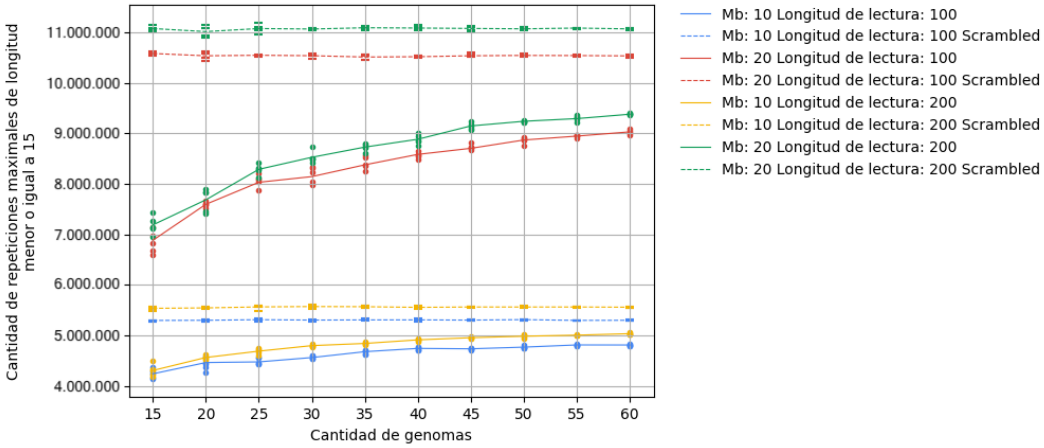

Figura 3.9: Cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para conjuntos de metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

En la *Figura* 3.9 vemos los resultados de la Propiedad 3. Como era esperado, obtuvimos un gráfico muy similar al de la Propiedad 1. Nuevamente podemos observar que los gráficos correspondientes a los metagenomas crecen a medida que aumenta la cantidad de genomas mientras que los de los metagenomas scrambled se mantienen constantes. Por lo tanto, para esta propiedad también existe una correlación para los metagenomas pero no para los metagenomas scrambled. Esto se puede corroborar con los valores de Spearman y Pearson como se ve en la *Tabla* 3.7 y la *Tabla* 3.8:

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,9757	1,46e-06	1	6,64e-64	1	6,64e-64	1	6,64e-64
Metagenoma scrambled	0,0424	0,9073	-0,2969	0,4047	0,1515	0,676	0,2121	0,5563

Tabla 3.7: Valores del coeficiente de correlación de Spearman para la cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,9376	6,13e-05	0,9449	3,74e-05	0,9263	0,0001	0,9471	3,2e-05
Metagenoma scrambled	0,0608	0,8674	-0,4046	0,246	0,4741	0,1662	0,3715	0,2904

Tabla 3.8: Valores del coeficiente de correlación de Pearson para la cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Para la propiedad 3 obtuvimos resultados aún mejores que para las propiedades 1 y 2.

Propiedad 4: Valor del r máximo

En este caso analizamos la cantidad de repeticiones maximales correspondiente a la longitud con mayor cantidad de repeticiones maximales. Utilizamos la siguiente fórmula para el cálculo de la propiedad:

$$máximo(metagenoma) = r_i(metagenoma)$$
$$\text{para } r_i(metagenoma) \geq r_j(metagenoma)\, 1 \leq i, j \leq 20$$

Dado que graficaremos el r máximo, los valores de cada punto van a ser menores que los de la *Figura* 3.9, donde graficamos el valor de r_i para $1 \leq i \leq 15$. A continuación vemos, en la *Figura* 3.10, los resultados de la Propiedad 4.

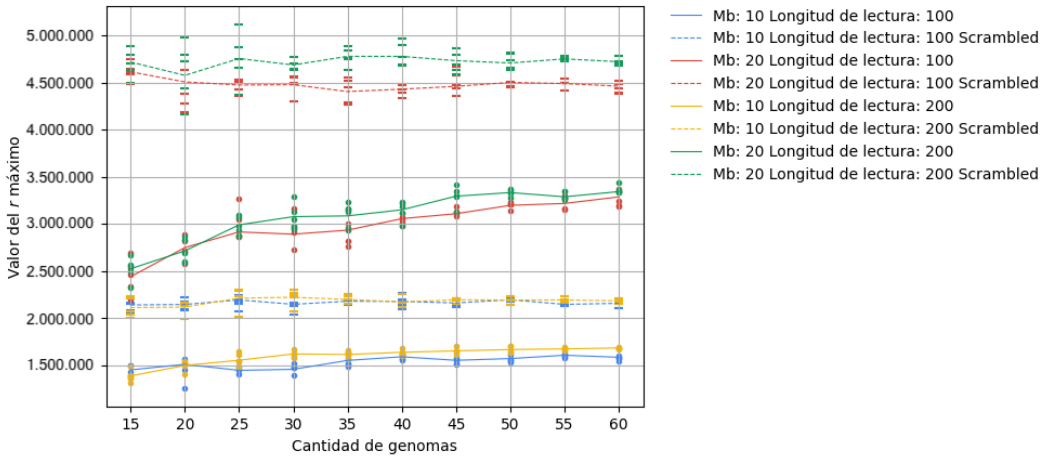

Figura 3.10: Cantidad de repeticiones maximales correspondiente a la longitud con mayor cantidad de repeticiones maximales en función de la cantidad de genomas para conjuntos de metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Como era esperado, en la *Figura* 3.10 vemos que todos los valores son menores en comparación con los de la *Figura* 3.9. Podemos observar un crecimiento solamente para los valores de los metagenomas, aunque de forma no tan evidente como en las propiedades anteriores ya que se observan oscilaciones en los resultados que antes no estaban presentes. Veamos los valores de Spearman y Pearson para esta propiedad en la *Tabla* 3.9 y la *Tabla* 3.10.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,8424	0,0022	0,9878	9,3e-08	0,9878	9,3e-08	0,9636	7,32e-06
Metagenoma scrambled	0,3212	0,3654	-0,3696	0,293	0,1272	0,726	0,2242	0,5334

Tabla 3.9: Valores del coeficiente de correlación de Spearman para el valor del r máximo en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,8440	0,0021	0,9462	3,41e-05	0,8951	0,0004	0,9236	0,0001
Metagenoma scrambled	0,1866	0,6056	-0,4514	0,1903	0,48	0,1602	0,3593	0,3077

Tabla 3.10: Valores del coeficiente de correlación de Pearson para el valor del r máximo en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Analizando los valores de las tablas anteriores podemos ver que si bien existe una correlación, esta es más débil en comparación con las 3 propiedades anteriores. Esto es esperable debido a las oscilaciones vistas en la *Figura* 3.10.

Propiedad 5: Diferencia entre el r máximo y el r máximo desplazado 5 posiciones

En la *Figura* 3.8 podemos ver que los gráficos llegan a un máximo y luego decrecen hasta casi no presentar repeticiones maximales. Por eso, analizamos la propiedad de la diferencia entre el r máximo y el r máximo desplazado en 5 unidades, donde todavía encontramos una cantidad de repeticiones maximales significativa. Esta propiedad se calcula de la siguiente forma:

$$diferencia\_máximo\_y\_máximo\_desplazado\_5(metagenoma)$$
$$= r_i(metagenoma) - r_{i+5}(metagenoma)$$
$$\text{para } r_i(metagenoma) \geq r_j(metagenoma)\, 1 \leq i, j \leq 20$$

En este caso obtendremos valores menores a los de la *Figura* 3.10, donde graficamos el r máximo, ya que graficaremos el r máximo restándole el valor de su desplazamiento en 5 unidades. En la *Figura* 3.11 podemos ver los resultados obtenidos para esta propiedad.

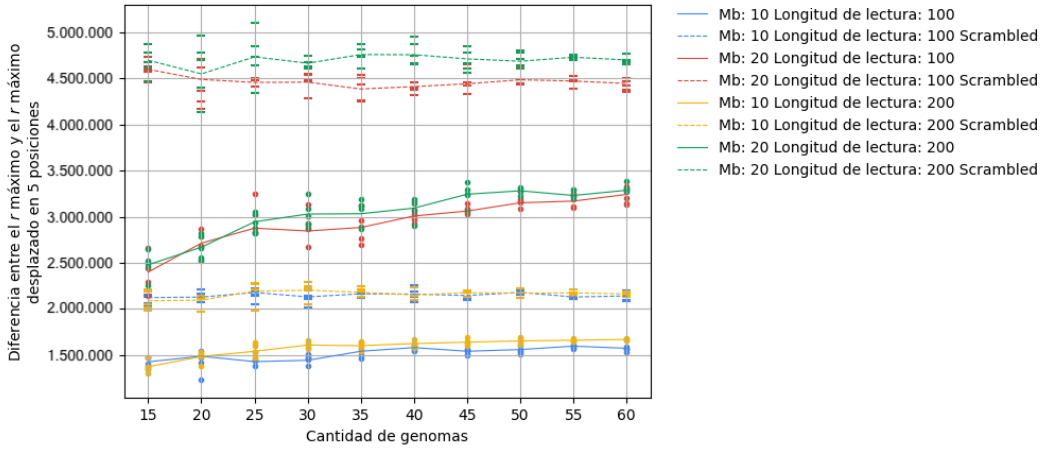

Figura 3.11: Diferencia entre el valor del r máximo y el r máximo desplazado en 5 posiciones en función de la cantidad de genomas para conjuntos de metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

En la *Figura* 3.11 se observa un gráfico muy similar al de la propiedad anterior y nuevamente existen oscilaciones en los valores obtenidos. Como era esperado, obtuvimos valores menores en comparación a de la *Figura* 3.11, aunque la diferencia no es significativa por lo que el valor que restamos parece ser muy bajo. Veamos qué sucede con los coeficientes de correlación:

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,8424	0,0022	0,9878	9,3e-08	0,9878	9,3e-08	0,9636	7,32e-06
Metagenoma scrambled	0,3212	0,3654	-0,3696	0,293	0,1272	0,726	0,2242	0,5334

Tabla 3.11: Valores del coeficiente de correlación de Spearman para el valor del r máximo y el r máximo desplazado en 5 posiciones en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,8583	0,0014	0,9446	3,84e-05	0,8922	0,0005	0,9207	0,0001
Metagenoma scrambled	0,1887	0,6014	-0,4483	0,1937	0,4837	0,1566	0,3614	0,3047

Tabla 3.12: Valores del coeficiente de correlación de Pearson para el valor del r máximo y el r máximo desplazado en 5 posiciones en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

La *Tabla* 3.11 y la *Tabla* 3.12 vuelven a mostrar una correlación entre la propiedad evaluada y la cantidad de genomas.

Propiedad 6: Diferencia entre r_{12} y r_{17}

En este caso analizamos la diferencia entre r_{12} y r_{17}. Esta propiedad es similar a la anterior por lo que esperamos resultados parecidos. En lugar de calcular la diferencia entre el r máximo y el r máximo desplazado en 5 unidades, lo hacemos desde r_{12} ya que en todos los gráficos encontramos el máximo en r_{11} o r_{12}. Al fijar el primer valor en r_{12} y desplazarnos 5 unidades calcularemos su diferencia con r_{17}, utilizando la siguiente fórmula:

$$\begin{aligned} &diferencia\_r\_12\_y\_r\_17(metagenoma) \\ &= r_{12}(metagenoma) - r_{12+5}(metagenoma) \\ &= r_{12}(metagenoma) - r_{17}(metagenoma) \end{aligned}$$

A continuación vemos los resultados de esta propiedad:

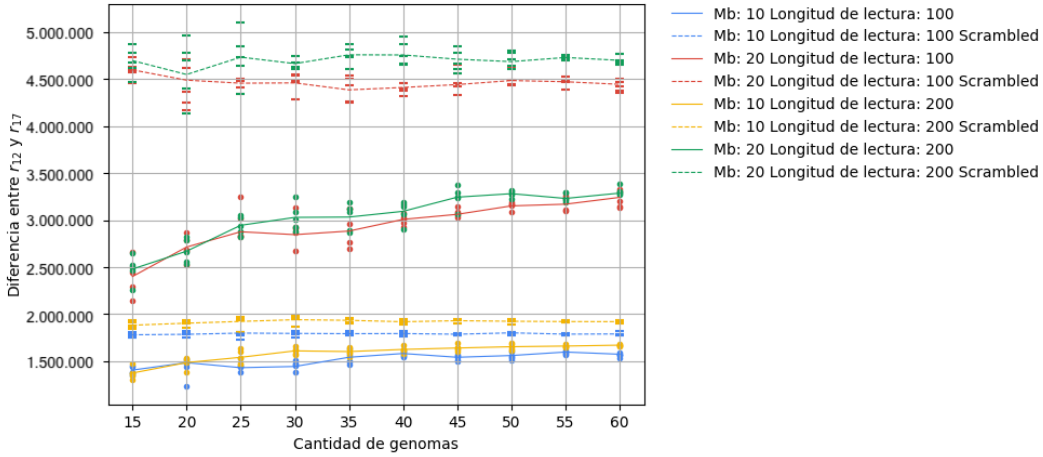

Figura 3.12: Diferencia entre r_{12} y r_{17} en función de la cantidad de genomas para conjuntos de metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

La *Figura* 3.12 muestra resultados similares a las propiedades anteriores, pero sobre todo a la Propiedad 5. Todos los valores se mantienen iguales con excepción de los metagenomas scrambled de tamaño 10, que ahora presentan valores menores. Esto es esperable ya que, como se ve en la *Figura* 3.8, el máximo en estos casos es r_{11} y no r_{12}. Para esta propiedad nuevamente se observan oscilaciones en sus valores.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,8424	0,0022	0,9878	9,3e-08	0,9878	9,3e-08	0,9636	7,32e-06
Metagenoma scrambled	0,2484	0,4887	-0,3696	0,2930	0,2484	0,4887	0,2242	0,5334

Tabla 3.13: Valores del coeficiente de correlación de Spearman para la diferencia entre r_{12} y r_{17} en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

	Mb: 10 Longitud: 100		Mb: 20 Longitud: 100		Mb: 10 Longitud: 200		Mb: 20 Longitud: 200	
	valor	p-valor	valor	p-valor	valor	p-valor	valor	p-valor
Metagenoma	0,8699	0,001	0,9446	3,84e-05	0,8922	0,0005	0,9207	0,0001
Metagenoma scrambled	0,2412	0,5020	-0,4483	0,1937	0,4705	0,1698	0,3614	0,3047

Tabla 3.14: Valores del coeficiente de correlación de Pearson para la diferencia entre r_{12} y r_{17} en función de la cantidad de genomas para metagenomas y metagenomas scrambled generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Analizando los valores de la *Tabla* 3.13 y la *Tabla* 3.14 vemos que existe una correlación. Los valores son similares entre la Propiedad 5 y la Propiedad 6, siendo mejores los de la primera.

3.3. Correlación con cantidad de genomas

Como se pudo observar, para todas la propiedades anteriores calculamos los coeficientes de Spearman y Pearson y evaluamos su correlación con la cantidad de genomas. Obtuvimos

los mejores resultados al evaluar las primeras 3 propiedades siendo todos los coeficientes de Spearman muy similares entre sí y los coeficientes de Pearson ligeramente mejores para las propiedades 1 y 3. Decidimos elegir la Propiedad 3: Cantidad de repeticiones maximales de longitud hasta 15 sobre la Propiedad 1: Cantidad de repeticiones maximales, dado que en los histogramas observamos que la cantidad de repeticiones maximales es mas abundante en ese rango. Utilizaremos esta propiedad para intentar obtener un método para la estimación de la diversidad.

4. GENERACIÓN DE MODELO PREDICTIVO

En el capítulo anterior vimos mediante los coeficientes de Spearman y Pearson que existe una correlación entre la cantidad de genomas y la Propiedad 3: Cantidad de repeticiones maximales de longitud hasta 15. Utilizaremos esta propiedad para proponer un método que nos permita estimar la diversidad de un metagenoma. Para esto evaluaremos la propiedad para metagenomas generados con distinta cantidad de genomas y luego buscaremos aproximar estos resultados mediante una función. Esta función devolverá una aproximación de la propiedad a partir de la cantidad de genomas. Finalmente, calcularemos la función *metadiversidad* como la inversa de la función de aproximación y la utilizaremos para obtener la cantidad de genomas a partir del cálculo de la propiedad elegida.

4.1. Función de metadiversidad

En el capítulo anterior seleccionamos la siguiente propiedad:

$$cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15(metagenoma) = \sum_{i=1}^{15} r_i(metagenoma)$$

donde $r_i(metagenoma)$ es la cantidad de repeticiones maximales de longitud i del metagenoma. Buscaremos una función que permita aproximar la propiedad anterior a partir de la cantidad de genomas del mismo. Para eso generaremos metagenomas variando la cantidad de genomas y fijando el tamaño t y la longitud de lecturas l, y calcularemos el valor de la Propiedad 3 para ellos. A partir de estos resultados evaluaremos distintas curvas para encontrar la que mejor se ajuste y de esta forma obtendremos la función:

$$aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_t\_l(cantidad\_de\_genomas(metagenoma)) \approx \sum_{i=1}^{15} r_i(metagenoma)$$

Esta función toma la cantidad de genomas de un metagenoma de tamaño t y longitud de lecturas l (estos son datos del metagenoma que siempre se suelen saber), y devuelve una aproximación del valor de la Propiedad 3 para el mismo. Por lo tanto, podemos utilizar su inversa para obtener una estimación de la cantidad de genomas a partir del cálculo de la propiedad. Llamaremos *metadiversidad* a la inversa de esta función:

$$metadiversidad\_t\_l(metagenoma)$$
$$= aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_t\_l^{-1}($$
$$\sum_{i=1}^{15} r_i(metagenoma))$$

Si calculamos las repeticiones maximales de un metagenoma de tamaño t y longitud de lecturas l, calculamos el valor de la propiedad elegida y evaluamos ese resultado en la función $metadiversidad\_t\_l(metagenoma)$, hipotetizamos que obtendremos una estimación de la cantidad de genomas del metagenoma. En la *Figura* 4.1 podemos ver los pasos para la estimación de la diversidad de un metagenoma.

4.2. Conjuntos de entrenamiento y testeo

Separamos los genomas en dos conjuntos, uno de entrenamiento y otro de testeo. Ambos conjuntos pueden verse en la sección B del apéndice. Utilizaremos el conjunto de entrenamiento para generar las funciones $metadiversidad\_t\_l$ y el conjunto de testeo para evaluarla.

Para armar los conjuntos de entrenamiento y de testeo separamos los 80 genomas en dos conjuntos de 40 genomas cada uno, de forma que los conjuntos sean similares entre sí. Para lograr esto, construímos los conjuntos de manera que los tamaños y el contenido GC se distribuyan de forma similar entre ambos. El contenido GC (contenido de guanina y citosina) se calcula como la cantidad de guanina y citosina sobre la cantidad total de nucleótidos, es decir:

$$\text{Contenido GC} = \frac{G + C}{A + T + C + G}$$

En la *Figura* 4.2 vemos el tamaño de los genomas de cada conjunto ordenados de menor a mayor. En azul se ve el conjunto de entrenamiento y en rojo el de testeo. Podemos ver que los tamaños de los genomas de ambos conjuntos tienen una distribución similar. La *Figura* 4.3 muestra un gráfico igual al anterior pero para el contenido GC. También concluímos a partir del mismo que ambos conjuntos tienen valores similares para el contenido GC de sus genomas.

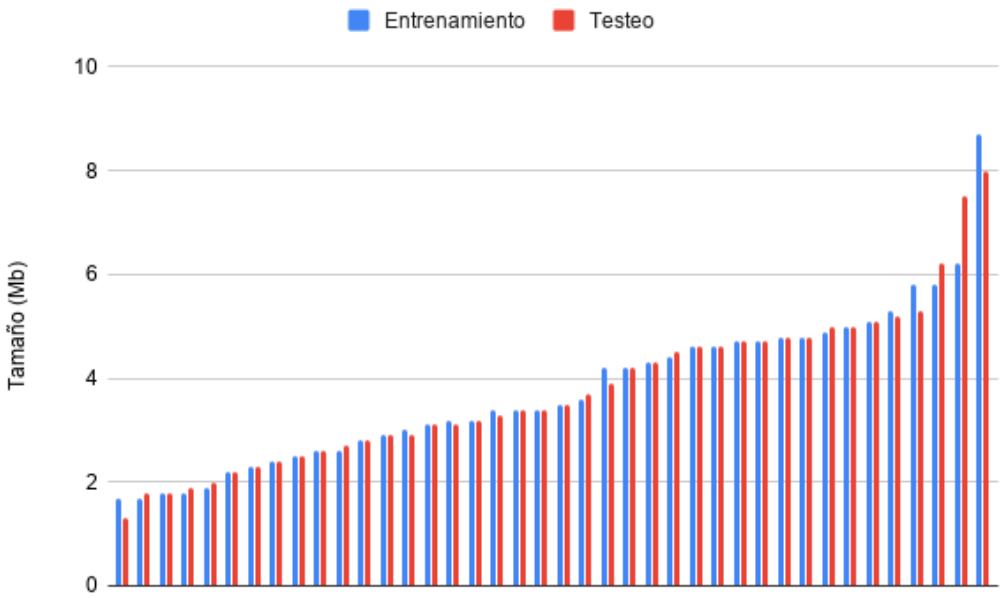

Figura 4.1: Tamaño en Mb para los conjuntos de genomas de entrenamiento y testeo ordenados de menor a mayor tamaño.

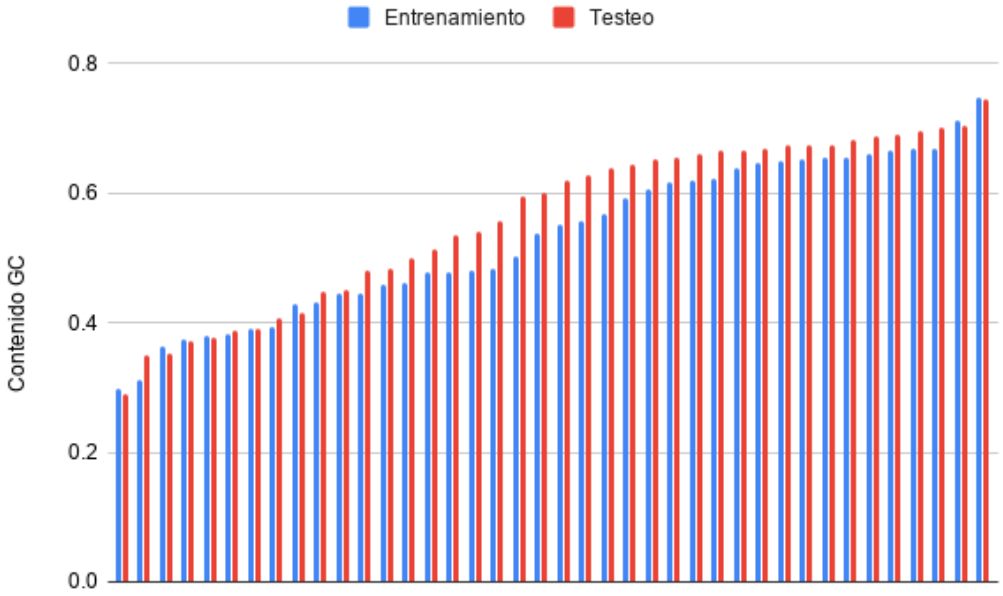

Figura 4.2: Contenido GC para los conjuntos de genomas de entrenamiento y testeo ordenados de menor a mayor contenido GC.

4.3. Función de aproximación

Queremos encontrar la función:

$$aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_t\_l(cantidad\_de\_genomas(metagenoma))$$

que a partir de la cantidad de genomas de un metagenoma nos permita obtener una aproximación del valor de la Propiedad 3. Para esto, calculamos el valor de la propiedad para metagenomas generados a partir del conjunto de entrenamiento con diferentes cantidades de genomas y fijando el tamaño y la longitud de lecturas. Una vez obtenidos estos valores intentamos ajustar la curva con distintas funciones para encontrar la que mejor se aproxime. Por ejemplo, para un conjunto de metagenomas de 10 Mb y lecturas de longitud 100, obtuvimos las curvas de aproximación que pueden verse en las *Figura* 4.4, *Figura* 4.5, *Figura* 4.6 y *Figura* 4.7.

Utilizaremos el coeficiente R^2 para evaluar qué función se aproxima mejor. Este puede tomar valores entre 0 y 1 indicando una mejor aproximación a medida que los valores se acercan a 1. En la *Tabla* 4.1 vemos los valores del R^2 para cada una de las funciones evaluadas.

	Función lineal	Polinomio de grado 2	Logaritmo natural	Exponencial
R^2	0,887	**0,972**	0,961	0,875

Tabla 4.1: Valores del coeficiente R^2 para distintas curvas que aproximan la cantidad de repeticiones maximales de longitud hasta 15 para metagenomas de longitud de lectura 100 y tamaño 10 Mb.

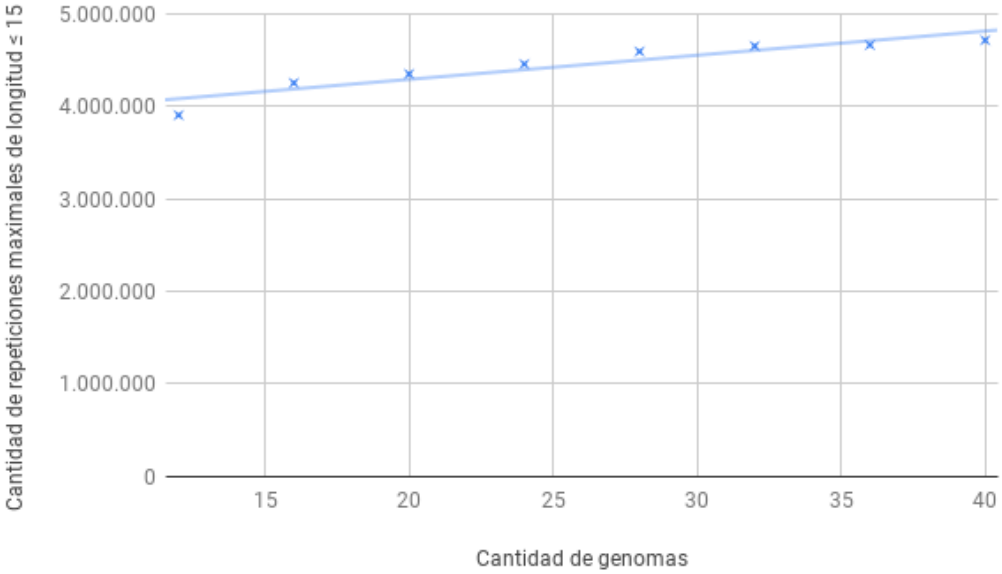

Figura 4.3: Ajuste con una función lineal de la cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para metagenoma generado de 10 Mb y lecturas de longitud 100.

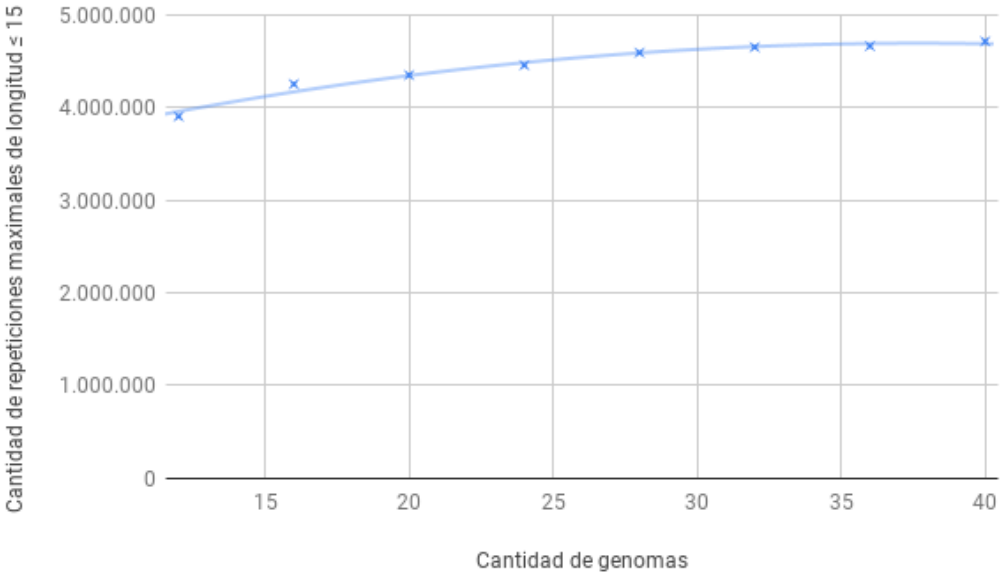

Figura 4.4: Ajuste con un polinomio de grado 2 de la cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para metagenoma generado de 10 Mb y lecturas de longitud 100.

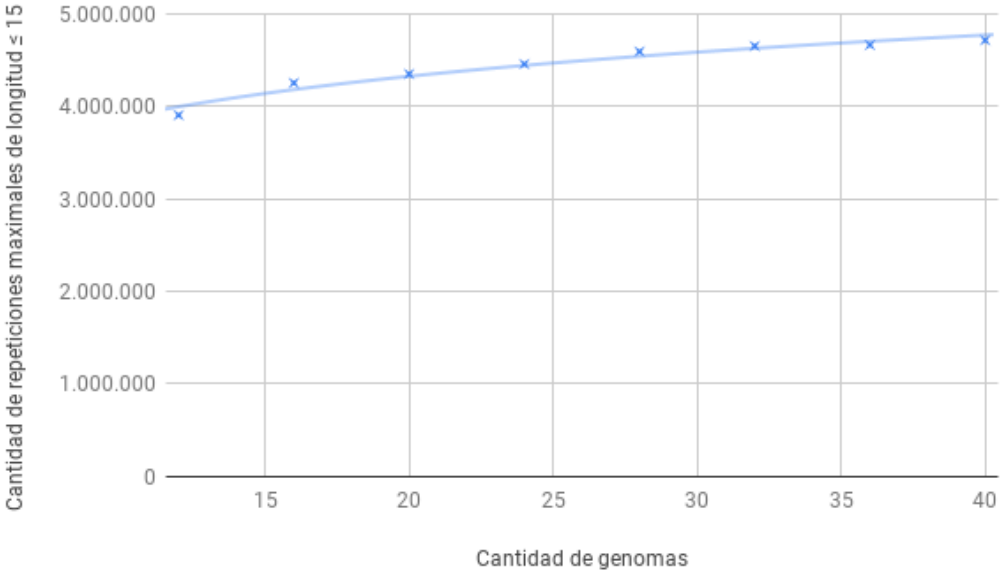

Figura 4.5: Ajuste con una función logarítmica de la cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para metagenoma generado de 10 Mb y lecturas de longitud 100.

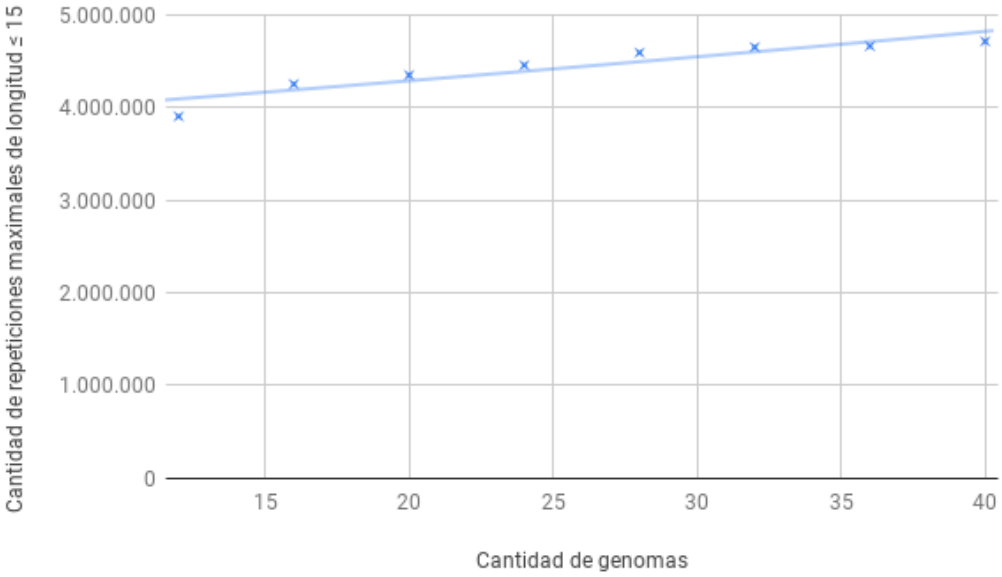

Figura 4.6: Ajuste con una función exponencial de la cantidad de repeticiones maximales de longitud hasta 15 en función de la cantidad de genomas para metagenoma generado de 10 Mb y lecturas de longitud 100.

Todos los R^2 en la *Tabla* 4.1 son cercanos a 1 como era esperado por lo observado en los gráficos, donde podemos ver que todas las curvas de ajuste se acercan a los puntos. El mejor coeficiente lo obtenemos para el polinomio de grado 2.

Aplicamos el mismo procedimiento para metagenomas generados con distintos tamaños y longitudes de lectura. En la *Tabla* 4.2 vemos los resultados del R^2 para los distintos parámetros explorados:

Tamaño(Mb)	Longitud de lectura	Función lineal	Polinomio de grado 2	Logaritmo natural	Exponencial
10	100	0,887	**0,972**	**0,961**	0,875
20	100	0,853	**0,946**	**0,937**	0,833
10	200	0,926	**0,983**	**0,98**	0,915
20	200	0,938	**0,991**	**0,989**	0,921

Tabla 4.2: Valores del coeficiente R^2 para distintas curvas que aproximan la cantidad de repeticiones maximales de longitud hasta 15 para metagenomas generados con distintas combinaciones de tamaño de archivo y de longitud de lectura.

Para todas las combinaciones de parámetros obtuvimos que la función que mejor se aproxima es el polinomio de grado 2 seguido por el logaritmo natural. Sin embargo, la inversa de un polinomio de grado 2 tiene una raíz cuadrada por lo que al intentar utilizarla tuvimos que evaluar valores que estaban fuera del dominio. Adicionalmente, el polinomio tiene 3 parámetros y el logaritmo tiene solamente 2 haciendo que sea preferente el uso de este último.

Dado que el R^2 del logaritmo es muy similar al de la función cuadrática y, por los

motivos explicados anteriormente, decidimos utilizar el logaritmo natural como función de aproximación. Estas son las 4 funciones de aproximación obtenidas:

$$aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_10\_100(cantidad\_de\_genomas(metagenoma)) = 564{,}761 \ln(cantidad\_de\_genomas(metagenoma)) + 2{,}660{,}000$$

$$aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_10\_200(cantidad\_de\_genomas(metagenoma)) = 650{,}509 \ln(cantidad\_de\_genomas(metagenoma)) + 2{,}560{,}000$$

$$aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_20\_100(cantidad\_de\_genomas(metagenoma)) = 1{,}740{,}000 \ln(cantidad\_de\_genomas(metagenoma)) + 2{,}330{,}000$$

$$aproximación\_cantidad\_repeticiones\_maximales\_de\_longitud\_hasta\_15\_20\_200(cantidad\_de\_genomas(metagenoma)) = 1{,}780{,}000 \ln(cantidad\_de\_genomas(metagenoma)) + 2{,}480{,}000$$

A partir de la función de aproximación calcularemos la función *metadiversidad* como su inversa. A partir de las 4 funciones de aproximación anteriores obtenemos 4 funciones de metadiversidad:

$$metadiversidad\_10\_100(metagenoma) = e^{\frac{\sum_{i=1}^{15} r_i(metagenoma) - 2{,}660{,}000}{564{,}761}}$$

$$metadiversidad\_10\_200(metagenoma) = e^{\frac{\sum_{i=1}^{15} r_i(metagenoma) - 2{,}330{,}000}{1{,}740{,}000}}$$

$$metadiversidad\_20\_100(metagenoma) = e^{\frac{\sum_{i=1}^{15} r_i(metagenoma) - 2{,}560{,}000}{650{,}509}}$$

$$metadiversidad\_20\_200(metagenoma) = e^{\frac{\sum_{i=1}^{15} r_i(metagenoma) - 2{,}480{,}000}{1{,}780{,}000}}$$

Para poder estimar la diversidad de un metagenoma mediante estas funciones y obtener valores razonables necesitamos que este cumpla con los parámetros seleccionados para la generación de las mismas. Es decir, utilizaremos la función *metadiversidad*_10_100 solamente para estimar la diversidad en metagenomas con 10 Mb de tamaño y lecturas de longitud 100. Dado que queremos utilizar estas funciones para cualquier metagenoma,

tendremos que modificarlo para que cumpla con los parámetros utilizados para generar la función. Para estimar la diversidad de un metagenoma tomaremos una submuestra del mismo que cumpla con alguna de la combinación de parámetros utilizada y la evaluaremos en la función de metadiversidad correspondiente.

4.4. Algoritmo

A continuación presentaremos el algoritmo para el cálculo de la metadiversidad junto con un análisis del orden algorítmico del mismo.

4.4.1. Algoritmo para el cálculo de la metadiversidad

En el *Algoritmo* 4.1 vemos un pseudocódigo del algoritmo utilizado para evaluar un metagenoma en una de las funciones de metadiversidad:

```
1. metadiversity(metagenome, t, l):
2.    sample ← sample(metagenome, t, l)
3.    concat ← empty string
4.    for read in sample do:
5.       concat ← concat + # + read
6.    maxRep ← maxRepeat(concat)
7.    suma_hasta_15 ← 0
8.    for i from 1 to length(maxRep) do:
9.       length ← i - maxRep[i]
10.      if length ≤ 15:
11.          suma_hasta_15 ← suma_hasta_15 + 1
12.   if t == 10 and l == 100:
13.      return metadiversidad_10_100(suma_hasta_15)
14.   if t == 10 and l == 200:
15.      return metadiversidad_10_200(suma_hasta_15)
16.   if t == 20 and l == 100:
17.      return metadiversidad_20_100(suma_hasta_15)
18.   if t == 20 and l == 200:
19.      return metadiversidad_20_200(suma_hasta_15)
```

Algoritmo 4.1: Algoritmo para la estimación de la diversidad en metagenomas simulados

En la línea 2 se toma una submuestra del metagenoma para que cumpla con los parámetros necesarios por la función de metadiversidad. Luego se calculan las repeticiones maximales de la submuestra. Desde la línea 3 a la 5 se concatenan todas las lecturas del metagenoma y en la línea 6 se obtienen todas las repeticiones maximales utilizando la modificación realizada al algoritmo de Ilie. A partir del arreglo *maxRep* obtenido utilizando la adaptación del algoritmo de Ilie (*Algoritmo* 2.1), se calcula la cantidad de repeticiones maximales de longitud menor o igual a 15. Esto se hace desde la línea 8 hasta la 11. Finalmente se evalúa el resultado de la propiedad en la función de metadiversidad apropiada según los parámetros elegidos.

Notar que *metagenome* tiene que tener un tamaño de al menos t Mb y lecturas de l o más nucleótidos.

4.4.2. Orden algorítmico

Tomar una muestra del metagenoma tiene complejidad de $O(n)$ siendo n la longitud total del metagenoma. Concatenar todas las lecturas del metagenomas y evaluar la nueva cadena de caracteres en el algoritmo de Ilie también tiene complejidad $O(n)$ como se explicó anteriormente. Para obtener la cantidad de repeticiones maximales de longitud menor o igual a 15 alcanza con recorrer el arreglo $maxRep$ una única vez siendo esto también $O(n)$. Finalmente calculamos el valor de la función de metadiversidad que equivale a calcular un logaritmo, con complejidad $O(1)$. Por lo tanto, estimar la diversidad de un metagenoma con una de las funciones de diversidad tiene una complejidad de $O(n)$. En este tipo de aplicaciones que procesan grandes cantidades de datos es muy bueno desarrollar algoritmos con órdenes temporales lineales.

4.5. Evaluación del método

Evaluaremos el método propuesto calculando la diversidad de metagenomas generados a partir de los genomas del conjunto de testeo. Para cada metagenoma simulado estimaremos la diversidad utilizando la función de metadiversidad correspondiente según los parámetros elegidos para su generación. Finalmente calcularemos el error absoluto y el error relativo de los resultados obtenidos y del promedio de los resultados.

$$\text{error absoluto} = |\textit{genomas} - \textit{estimación diversidad}|$$

$$\text{error relativo} = \frac{|\textit{genomas} - \textit{estimación diversidad}|}{\textit{genomas}} = \frac{\textit{error absoluto}}{\textit{genomas}}$$

Armamos conjuntos de 12, 16, 20, 24, 28, 32, 36 y 40 genomas seleccionados del conjunto de testeo. Simulamos conjuntos utilizando tamaños a partir de 12 ya que las estimaciones con una cantidad menor pueden tener un error alto, y aumentando los valores en 4 unidades para disminuir la cantidad de ejecuciones del experimento. Para cada uno de estos conjuntos generamos 4 metagenomas combinando las longitudes de lectura de 100 y 200 y los tamaños de archivo de 10 y 20 Mb. En la *Tabla* 4.3 podemos ver la diversidad obtenida para cada uno de los metagenomas de testeo generados según los parámetros utilizados para la generación y su promedio.

Cantidad de genomas	Mb: 10 Longitud: 100	Mb: 10 Longitud: 200	Mb: 20 Longitud: 100	Mb: 20 Longitud: 200	Promedio
12	9,0714	9,5787	9,5713	9,0446	9,3165
16	16,8315	17,0663	15,7021	15,2017	16,2004
20	19,9684	20,4149	18,4899	18,2403	19,2784
24	24,1509	24,0521	23,5134	23,4793	23,7989
28	30,7064	30,138	29,0022	29,1731	29,7549
32	34,1327	33,6795	32,4719	32,906	33,2975
36	34,8342	34,2341	34,1091	35,2346	34,603
40	38,1301	37,9142	37,2941	38,1587	37,8743

Tabla 4.3: Diversidad estimada para metagenomas generados con distintos tamaños y longitudes de lectura.

Para evaluar los resultados obtenidos calculamos el error absoluto y el error relativo de todos los resultados y del promedio. En las tablas *Tabla* 4.4 y *Tabla* 4.5 vemos los errores y los errores relativos respectivamente.

Cantidad de genomas	Mb: 10 Longitud: 100	Mb: 10 Longitud: 200	Mb: 20 Longitud: 100	Mb: 20 Longitud: 200	Promedio
12	2,928	2,421	2,428	2,955	2,683
16	0,831	1,066	0,297	0,798	0,2
20	0,031	0,414	1,51	1,759	0,721
24	0,15	0,052	0,486	0,52	0,201
28	2,706	2,138	1,002	1,173	1,754
32	2,132	1,679	0,471	0,906	1,297
36	1,165	1,765	1,89	0,765	1,396
40	1,869	2,085	2,705	1,841	2,125

Tabla 4.4: Error absoluto de la diversidad estimada para metagenomas generados con distintos tamaños y longitudes de lectura.

Cantidad de genomas	Mb: 10 Longitud: 100	Mb: 10 Longitud: 200	Mb: 20 Longitud: 100	Mb: 20 Longitud: 200	Promedio
12	0,244	0,201	0,202	0,246	0,223
16	0,051	0,066	0,018	0,049	0,012
20	0,001	0,02	0,075	0,087	0,036
24	0,006	0,002	0,02	0,021	0,008
28	0,096	0,076	0,035	0,041	0,062
32	0,066	0,052	0,014	0,028	0,04
36	0,032	0,049	0,052	0,021	0,038
40	0,046	0,052	0,067	0,046	0,053

Tabla 4.5: Error relativo de la diversidad estimada para metagenomas generados con distintos tamaños y longitudes de lectura.

En todos los casos obtuvimos errores bajos, observando un error de 1 orden de magnitud mayor al resto para el caso de 12 genomas. A partir del análisis de los errores obtenidos no encontramos una función que arroje mejores resultados que las otras generadas con otros parámetros. A diferencia de lo esperado, no observamos que el promedio tenga un error menor que las funciones de metadiversidad utilizadas de forma independiente.

4.6. Evaluación del método en metaviromas simulados

A continuación pondremos a prueba nuestro método aplicándolo al conjunto de datos del trabajo de Roux, Emerson, Eloe-Fadrosh, Sullivan [6] de manera de proveer una validación independiente de los datos usados para obtener el modelo. En el trabajo de Roux et al. se generaron 14 metagenomas utilizando genomas de virus, a diferencia de lo realizado en nuestro trabajo, donde utilizamos genomas de bacterias. Cada metagenoma se compone de entre 500 y 1000 genomas cada uno, mientras que nosotros usamos entre 12 y 40 genomas.

Otra diferencia está en el método de generación donde la cantidad de lecturas tomadas de cada genoma no se seleccionó de forma equiprobable.

En la *Figura* 4.8 vemos en el eje x los nombres de las muestras ordenados por cantidad de genomas. Los puntos negros muestran la cantidad de genomas utilizados para simular cada muestra.

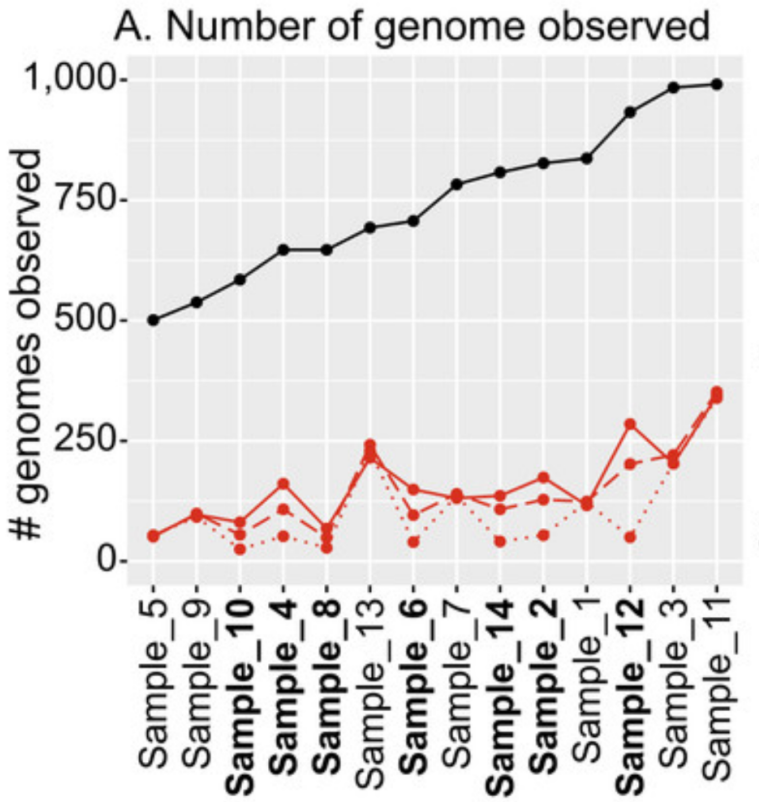

Figura 4.7: Figura extraída del trabajo de Roux et al. [6]. En negro: número de genomas utilizados para la generación de metagenomas. En rojo: cantidad de genomas observados.

Todos los metagenomas de Roux et al., tienen longitud de lectura 100 por lo que solamente evaluaremos las funciones $metadiversidad\_10\_100$ y $metadiversidad\_20\_100$.

Tomamos submuestras de cada metagenoma de manera de obtener uno de 10 Mb y otro de 20 Mb y evaluamos las submuestras en la función de metadiversidad correspondiente. Lamentablemente, todos los valores obtenidos fueron menores a 1. Hipotetizamos que esto puede deberse a que al tomar submuestras de 10 Mb y de 20 Mb, no llegamos a obtener información de todos los genomas presentes en las muestras. Recordemos que en nuestro caso obtuvimos errores bajos trabajando con solamente 40 genomas, mientras que los metagenomas del trabajo de Roux et al. fueron generados con entre 500 y 1000 genomas cada uno. Otra hipótesis es que esto puede deberse a que generamos las funciones de metadiversidad con genomas de bacterias mientras que esta prueba se realizó con genomas de virus.

Si bien no obtuvimos valores razonables de diversidad, ordenamos las muestras según la diversidad obtenida y lo comparamos con el orden original. En la *Tabla* 4.6 vemos las muestras ordenadas según su diversidad:

	Original	*metadiversidad*_10_100	*metadiversidad*_20_100
1	Sample_5	Sample_5	Sample_7
2	Sample_9	Sample_10	Sample_5
3	Sample_10	Sample_7	Sample_10
4	Sample_4	Sample_6	Sample_9
5	Sample_8	Sample_9	Sample_6
6	Sample_13	Sample_1	Sample_1
7	Sample_6	Sample_4	Sample_4
8	Sample_7	Sample_14	Sample_8
9	Sample_14	Sample_2	Sample_14
10	Sample_2	Sample_3	Sample_2
11	Sample_1	Sample_8	Sample_3
12	Sample_3	Sample_11	Sample_13
13	Sample_11	Sample_13	Sample_11

Tabla 4.6: Muestras generadas en el trabajo de Roux et al. [6]. En la primera columna las muestras están ordenadas según la cantidad de genomas utilizadas para su generación, en la segunda columna se ordenan según el valor de diversidad estimado utilizando la función *metadiversidad*_10_100 y en la tercera columna usando la función *metadiversidad*_20_100.

En la primera columna de la *Tabla* 4.6 tenemos el nombre de las muestras del trabajo de Roux et al. ordenadas según las cantidad de genomas utilizados para su generación. En la segunda tenemos las muestras ordenadas por la cantidad de genomas estimada utilizando la función *metadiversidad*_10_100 y en la última columna tenemos lo mismo para *metadiversidad*_20_100.

Para evaluar la correlación entre el orden original y los órdenes obtenidos utilizamos los coeficientes de Spearman y Pearson. Para cada uno de los tres órdenes identificamos las muestras con su número de muestra y los ordenamos como se ve en la *Tabla* 4.6. Luego evaluamos los coeficientes de correlación entre el orden original y los dos órdenes obtenidos.

	valor	p-valor
*metadiversidad*_10_100	0,5219	0,0672
*metadiversidad*_20_100	0,6208	0,0235

Tabla 4.7: Valores del coeficiente de correlación de Spearman para el orden de la diversidad estimada en metagenomas generados con con longitud de lectura 100 y tamaños de archivo de 10 Mb y 20 Mb.

	valor	p-valor
*metadiversidad*_10_100	0,5210	0,0672
*metadiversidad*_20_100	0,6208	0,0235

Tabla 4.8: Valores del coeficiente de correlación de Pearson para el orden de la diversidad estimada en metagenomas generados con con longitud de lectura 100 y tamaños de archivo de 10 Mb y 20 Mb.

Los valores de la *Tabla* 4.7 y *Tabla* 4.8 muestran que existe cierta correlación entre

los órdenes obtenidos y el orden original. Concluímos que si bien obtuvimos un error de escala muy alto al intentar estimar la diversidad, nuestro método nos permitió ordenar los metagenomas según su diversidad con una efectividad del 60 %.

5. CONCLUSIÓN

5.1. Conclusión

En este trabajo propusimos un método para la estimación de la diversidad de metagenomas basado en el cálculo de sus repeticiones maximales. Para esto utilizamos una adaptación del algoritmo de Ilie [12] que se encuentra en la tesis de Rago [17] para poder trabajar con múltiples cadenas de caracteres y aplicarlo a las lecturas de un metagenoma.

Generamos distintos metagenomas tomando subsecuencias al azar de genomas conocidos. A partir del cálculo de sus repeticiones maximales evaluamos distintas propiedades sobre las mismas y analizamos sus correlaciones con la cantidad de genomas presentes utilizando los coeficientes de Spearman y Pearson. Obtuvimos los mejores resultados para las propiedades relacionadas a la cantidad de instancias y de patrones de repetición maximal de un metagenoma. Luego de este análisis seleccionamos la propiedad de la cantidad de instancias de longitud menor o igual a 15, la cual utilizamos para encontrar los parámetros de una función que nos permitió estimar la diversidad para metagenomas. Finalmente propusimos un método de complejidad temporal y espacial lineal para estimar la diversidad de un metagenoma mediante el uso de las funciones de metadiversidad encontradas.

Evaluamos este método utilizándolo en un conjunto de metagenomas generados obteniendo errores bajos entre 0 y 3 genomas de diferencia para todas las funciones. No encontramos que una combinación de tamaño de archivo y longitud de lecturas específica haya permitido generar un función que arroje mejores resultados que las otras. Finalmente evaluamos el método utilizando un conjunto de datos generado en el trabajo de Roux et al. [6] a partir de genomas de virus. Observando los errores obtenidos concluímos que no pudimos estimar la diversidad pero los coeficientes de Spearman y Pearson indicaron que pudimos ordenar parcialmente los metagenomas según la misma. Esto pudo deberse a que los datos utilizados se generaron a partir de virus mientras que nosotros utilizamos bacterias o a que en el trabajo de Roux et al. se utilizan entre 500 y 1000 virus mientras que nosotros utilizamos solamente 40 bacterias.

5.2. Trabajo futuro

En este trabajo utilizamos repeticiones maximales para la búsqueda de propiedades. El mismo trabajo podría realizarse utilizando otro tipo de repeticiones, por ejemplo, las repeticiones NE y SNE también mencionadas en el trabajo de Ilie. [12]

Para la estimación de la diversidad generamos metagenomas artificiales a partir de tomar al azar subsecuencias de genomas conocidos. La probabilidad de tomar una subsecuencia de un genoma en particular fue la misma para todos. Proponemos como trabajo futuro realizar el mismo análisis que en este trabajo utilizando metagenomas de distintas complejidades.

Adicionalmente proponemos utilizar una mayor cantidad de genomas para generar las funciones de metadiversidad y volver a evaluar las funciones de aproximación para los casos con mayor cantidad de genomas involucrados.

Apéndice A

APÉNDICE

A.1. Genomas utilizados

1. Nitrobacter winogradskyi Nb-255
2. Pelobacter carbinolicus DSM 2380
3. Moorella thermoacetica ATCC 3907
4. Saccharophagus degradans 2-40
5. Pseudomonas putida F1
6. Exiguobacterium 255-15
7. Chlorobium limicola DSMZ 245(T)
8. Psychrobacter arcticus 273-4
9. Dechloromonas aromatica RCB
10. Thermobifida fusca YX
11. Prochlorococcus sp. NATL2A
12. Ehrlichia canis Jake
13. Methanosarcina barkeri Fusaro
14. Thiobacillus denitrificans ATCC 2525
15. Anabaena variabilis ATCC 29413
16. Nitrosococcus oceani
17. Pseudomonas fluorescens Pf0-1
18. Rhodobacter sphaeroides 2.4.1
19. Geobacter metallireducens GS-15
20. Thiomicrospira crunogena XCL-2
21. Prochlorococcus marinus str. MIT 9312
22. Nitrosospira multiformis ATCC 25196
23. Rhodospirillum rubrum ATCC 11170
24. Anaeromyxobacter dehalogenans 2CP-C
25. Rhodopseudomonas palustris HaA2
26. Novosphingobium aromaticivorans DSM 12444 (F199)
27. Methanospirillum hungatei JF-1
28. Jannaschia sp. CCS1
29. Rhodopseudomonas palustris BisB18
30. Methylobacillus flagellatus strain KT
31. Polaromonas sp. JS666
32. Burkholderia xenovorans LB400
33. Methanococcoides burtonii DSM6242
34. Rhodopseudomonas palustris BisB5
35. Chromohalobacter salexigens DSM3043
36. Nitrobacter hamburgensis
37. Deinococcus geothermalis DSM11300
38. Sphingopyxis alaskensis RB2256
39. Burkholderia cenocepacia AU 1054
40. Rubrobacter xylanophilus DSM 9941
41. Pseudoalteromonas atlantica T6c
42. Cytophaga hutchinsonii ATCC 33406

43. Haemophilus somnus 129PT
44. Trichodesmium erythraeum IMS101
45. Shewanella sp. MR-7
46. Alkalilimnicola ehrlichei MLHE-1
47. Nitrosomonas eutropha C71
48. Shewanella frigidimarina NCMB400
49. Burkholderia ambifaria AMMD
50. Rhodopseudomonas palustris BisA53
51. Lactobacillus brevis ATCC367
52. Pediococcus pentosaceus ATCC25745
53. Oenococcus oeni PSU-1
54. Lactobacillus gasseri ATCC33323
55. Leuconostoc mesenteroides ATCC 8293
56. Streptococcus thermophilus LMD-9
57. Arthrobacter sp. FB24
58. Burkholderia cenocepacia HI2424
59. Syntrophobacter fumaroxidans MPOB
60. Magnetococcus sp. MC-1
61. Shewanella sp. ANA-3
62. Chlorobium phaeobacteroides DSMZ 266(T)
63. Paracoccus denitrificans PD1222
64. Nocardioides sp. JS614
65. Shewanella amazonensis
66. Shewanella sp. W3-18-1
67. Clostridium thermocellum ATCC 27405
68. Shewanella baltica OS155
69. Bradyrhizobium sp. BTAi1
70. Clostridium beijerinckii NCIMB 8052
71. Actinobacillus succinogenes 130Z
72. Kineococcus radiotolerans SRS30216
73. Frankia sp. EAN1pec
74. Chloroflexus aurantiacus J-10-fl
75. Bifidobacterium longum DJO10A
76. Prosthecochloris aestuarii SK413/DSMZ 271(t)
77. Pelodictyon phaeoclathratiforme BU-1 (DSMZ 5477(T))
78. Desulfitobacterium hafniense DCB-2
79. Enterococcus faecium DO
80. Ferroplasma acidarmanus fer1

A.2. Conjuntos de entrenamiento y testeo

En la *Tabla* A.1 y la *Tabla* A.2 vemos los genomas seleccionados para el conjunto de entrenamiento y de testeo. Para cada uno vemos el tamaño, la cantidad de nucleótidos A, T, C y G y el valor del contenido GC.

Conjunto de entrenamiento

Genoma	Tamaño(Mb)	A	T	C	G	CG
NC_007577.1	1.7	586668	589086	266828	266624	0.3121051529
NC_008528.1	1.7	554622	551277	335649	338970	0.3788891772
NC_008525.1	1.8	573425	574424	342144	342399	0.3735789067

NC_008532.1	1.8	569652	561226	362290	363201	0.3908118483
NC_021592.1	1.9	618692	610816	349928	355763	0.3646606886
NC_008497.1	2.2	614454	617792	544757	514222	0.4621890037
NC_007520.2	2.3	692379	688275	518310	528773	0.4313000131
NC_008025.1	2.4	410555	412598	823307	820746	0.6663622738
CP000232.1	2.5	582555	579592	732127	734515	0.5579154508
NC_007204.1	2.6	757045	759150	566354	568153	0.4280024688
NC_017960.1	2.6	834193	834579	512430	516936	0.3815097671
NC_007404.1	2.8	493826	493556	960626	961807	0.6606718984
CP001022.1	2.9	793700	792598	721128	726711	0.4771831331
NC_008639.1	3	803010	815605	768814	746475	0.4835148109
NC_007614.1	3.1	734021	732559	859512	858156	0.5394265773
NC_008048.1	3.2	579938	574094	1096877	1094262	0.6550155433
NC_008060.1	3.2	547233	542093	1107539	1097700	0.6693566525
NC_007484.1	3.4	869487	860302	878101	873806	0.5031763256
NC_007794.1	3.4	622120	618992	1161091	1159382	0.651528182
NC_008390.1	3.4	590478	587860	1188789	1189420	0.6686848227
CP000142.2	3.5	821021	824461	1007951	1012461	0.5511375943
NC_007963.1	3.6	666454	667598	1174260	1188339	0.6391187591
NC_007643.1	4.2	752218	751774	1423846	1424992	0.6544794995
NC_007802.1	4.2	810457	816031	1331305	1360187	0.6233220163
NC_007964.1	4.3	844577	842684	1359293	1360414	0.6171379052
NC_007298.1	4.4	914415	919910	1333138	1333643	0.5924723835
NC_008322.1	4.6	1253991	1244597	1149246	1144777	0.4786582929
NC_008750.1	4.6	1303178	1303698	1049082	1052423	0.4463328265
NC_007355.1	4.7	1475708	1461758	953682	946261	0.3927604633
NC_007958.1	4.7	859367	862490	1580204	1590657	0.6480776125
NC_007760.1	4.8	626052	632068	1882376	1872987	0.7490527045
NC_008577.1	4.8	1293371	1289444	1194096	1195296	0.4805495829
CP000282.1	4.9	1369925	1369938	1160438	1157231	0.4582608672
NC_008228.1	5	1435270	1437497	1158708	1155532	0.4461609556
NC_010175.1	5.1	1135063	1142035	1491828	1489617	0.5669716878
NC_007925.1	5.3	967727	964478	1794950	1786690	0.6495721225
CP000712.1	5.8	1137800	1135215	1846910	1840040	0.6186194046
NC_009617.1	5.8	2121554	2087554	886686	904841	0.2985562361
NC_007492.2	6.2	1276547	1265274	1952604	1943981	0.6052095814
NC_009921.1	8.7	1293213	1297784	3190127	3200920	0.7115359266
	151.1	35185961	35112762	43017331	43019310	0.5503338387

Tabla A.1: Genomas del conjunto de entrenamiento

Conjunto de testeo

Genoma	Tamaño(Mb)	A	T	C	G	CG
NC_007354.1	1.3	467984	466243	191546	189258	0.2895779643
NC_007335.2	1.8	599178	596442	322988	324295	0.3512300973
NC_008530.1	1.8	618280	608138	330583	337364	0.3525967805

NC_008309.1	1.9	629633	631255	373612	373202	0.3719745261
NC_008531.1	2	634012	635700	380483	388206	0.3771039163
NC_009655.1	2.2	640349	637355	519463	522498	0.4491859816
NC_010816.1	2.3	472113	474734	712299	716648	0.6014608169
NC_011059.1	2.4	629305	624766	633029	625824	0.5009514812
NC_007955.1	2.5	760719	764779	520331	529204	0.4075811844
NC_008344.1	2.6	685693	685117	644087	646162	0.4848629812
CP001097.1	2.7	671353	673851	714101	703878	0.5131686899
NC_008686.1	2.8	477689	471939	952536	950119	0.6670638923
NC_007947.1	2.9	656439	659377	828961	826742	0.5571907836
NC_011060.1	2.9	788810	778332	730628	720469	0.4807760419
NC_007493.2	3.1	493825	494381	1104535	1095784	0.6900742506
NC_008148.1	3.1	476689	475662	1140018	1133380	0.7047659319
NC_008340.1	3.2	533859	529800	1105679	1106608	0.6753124136
CP000115.1	3.3	647647	643481	1058584	1052382	0.620490204
NC_007796.1	3.4	981407	962916	803497	796919	0.4514905047
NC_008542.1	3.4	579367	577936	1159748	1166852	0.6678142302
NC_007333.1	3.5	592034	591594	1228187	1230436	0.6750284371
NC_009012.1	3.7	1170580	1174351	751240	747135	0.3898661725
NC_007517.1	3.9	814227	804348	1194303	1184545	0.5950953902
NC_008700.1	4.2	999899	998737	1150247	1157260	0.5358639971
NC_008255.1	4.3	1359220	1351852	860544	861608	0.3884649185
NC_008541.1	4.5	813564	809163	1539452	1536768	0.6546615657
NC_008576.1	4.6	1086849	1076254	1293545	1262935	0.5416749743
NC_009664.2	4.6	607030	610165	1765139	1778850	0.7443503549
NC_007951.1	4.7	910429	912982	1533044	1539382	0.6275588832
NC_008345.1	4.7	1414471	1415931	1005981	1008876	0.4158409282
NC_008554.1	4.8	1005043	993616	1494489	1497104	0.5994873606
NC_008699.1	4.8	704517	708880	1784268	1788207	0.7165195978
NC_007948.1	5	979163	972376	1623617	1625109	0.6247231631
NC_009052.1	5	1377567	1376684	1188157	1184969	0.4628343108
NC_011830.1	5.1	1388107	1381467	1248468	1261094	0.4753736217
NC_007778.1	5.2	905976	904742	1763763	1757177	0.660383693
NC_008435.1	5.3	977787	979821	1773364	1774524	0.6444265875
NC_007413.1	6.2	1870419	1858539	1315721	1321054	0.4142138855
NC_008312.1	7.5	2551359	2552603	1320480	1325668	0.3414336055
NC_009485.1	8	1448170	1451072	2687087	2678361	0.6492013615
	151.1	35420762	35317381	42747804	42726856	0.5471680833

Tabla A.2: Genomas del conjunto de testeo

Bibliografía

[1] Richard J. Randle-Boggis, Thorunn Helgason, Melanie Sapp and Peter D. Ashton. (2016). Evaluating techniques for metagenome annotation using simulated sequence data. FEMS Microbiology Ecology, 92, 2016, w095.

[2] Stein, Lincoln. (2011). An Introduction to the Informatics of "Next-Generation" Sequencing. Current protocols in bioinformatics / editorial board, Andreas D. Baxevanis ... [et al.]. Chapter 11. Unit 11.1.. 10.1002/0471250953.bi1101s36.

[3] Oulas, A., Pavloudi, C., Polymenakou, P., Pavlopoulos, G. A., Papanikolaou, N., Kotoulas, G., ... Iliopoulos, I. (2015). Metagenomics: tools and insights for analyzing next-generation sequencing data derived from biodiversity studies. Bioinformatics and biology insights, 9, 75–88.

[4] Germán Bonilla-Rosso, Luis E. Eguiarte, David Romero, Michael Travisano and Valeria Souza. (2012). Understanding microbial community diversity metrics derived from metagenomes: performance evaluation using simulated data sets. FEMS microbiology ecology. 82. 37-49. 10.1111/j.1574-6941.2012.01405.x.

[5] Allen, H. K., Bunge, J., Foster, J. A., Bayles, D. O., and Stanton, T. B. (2013). Estimation of viral richness from shotgun metagenomes using a frequency count approach. Microbiome, 1(1), 5.

[6] Simon Roux, Joanne B. Emerson, Emiley A. Eloe-Fadrosh and Matthew B. Sullivan. (2017). Benchmarking viromics: an in silico evaluation of metagenome-enabled estimates of viral community composition and diversity. PeerJ. 5. e3817. 10.7717/peerj.3817.

[7] Taur, Y., Jenq, R. R., Perales, M. A., Littmann, E. R., Morjaria, S., Ling, L., ... Pamer, E. G. (2014). The effects of intestinal tract bacterial diversity on mortality following allogeneic hematopoietic stem cell transplantation. Blood, 124(7), 1174–1182.

[8] Scher, J. U., Ubeda, C., Artacho, A., Attur, M., Isaac, S., Reddy, S. M., ... Abramson, S. B. (2015). Decreased bacterial diversity characterizes the altered gut microbiota in patients with psoriatic arthritis, resembling dysbiosis in inflammatory bowel disease. Arthritis & rheumatology (Hoboken, N.J.), 67(1), 128–139.

[9] Ai, D., Huang, R., Wen, J., Li, C., Zhu, J., & Xia, L. C. (2017). Integrated metagenomic data analysis demonstrates that a loss of diversity in oral microbiota is associated with periodontitis. BMC genomics, 18(Suppl 1), 1041.

[10] Gao, W., Weng, J., Gao, Y., & Chen, X. (2013). Comparison of the vaginal microbiota diversity of women with and without human papillomavirus infection: a cross-sectional study. BMC infectious diseases, 13, 271.

[11] Sunagawa, S., DeSantis, T.Z., Piceno, Y.M., Brodie, E.L., DeSalvo, M.K., Voolstra, C.R., Weil, E., Andersen, G.L., and Medina, M. (2009). Bacterial diversity and White Plague Disease-associated community changes in the caribbean coral Montastraea faveolata. ISME J. 3, 512–521.

[12] Ilie, L. and Smyth, W. F. (2011). Minimum Unique Substrings and Maximum Repeats. Fundam. Inform.. 110. 183-195. 10.3233/FI-2011-536.

[13] Manber, U. and Myers, G. (1990). Suffix Arrays: a New Method for On-Line String Searches. SIAM Journal on Computing. 22. 319-327. 10.1137/0222058.

[14] Pang Ko and Srinivas Aluru. (2003). Space efficient linear time construction of suffix arrays. Proceedings of the 14th Annual Conference on Combinatorial Pattern Matching, CPM'03, pages 200–210, Berlin, Heidelberg, 2003. Springer-Verlag.

[15] Juha Kärkkäinen and Peter Sanders. (2003) Simple linear work suffix array construction. Proceedings of the 30th International Conference on Automata, Languages and Programming, ICALP'03, pages 943–955, Berlin, Heidelberg, 2003. Springer-Verla

[16] Ge Nong, Sen Zhang, and Wai Hong Chan. (2009). Linear time suffix array construction using d-critical substrings. In Proceedings of the 20th Annual Symposium on Combinatorial Pattern Matching, CPM '09, pages 54–67, Berlin, Heidelberg, 2009. Springer-Verlag.

[17] Rago, P. (2017). Patrones de repetición para clasificación e identificación de proteínas. Tesis de licenciatura del Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires.

[18] Konstantinos Mavromatis, Natalia Ivanova, Kerrie Barry, Harris Shapiro, Eugene Goltsman, Alice C McHardy, Isidore Rigoutsos, Asaf Salamov, Frank Korzeniewski, Miriam Land3, Alla Lapidus, Igor Grigoriev, Paul Richardson, Philip Hugenholtz & Nikos C Kyrpides. (2007). Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. Nature Methods, v.4, 495-500 (2007). 4.