Estudio del uso de la NCD para la inferencia de árboles filogenéticos

Tesis de Licenciatura

Alumnos:

David Vacca mvacca@dc.uba.ar

Diego Larralde dlarralde@dc.uba.ar

Director:

Martín Urtasun murtasun@dc.uba.ar

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

ABSTRACT **4**

INTRODUCCIÓN **5**

CONCEPTOS BÁSICOS DE BIOLOGÍA MOLECULAR 5
Secuencias biológicas *6*
FUNDAMENTOS DE LA DISTANCIA DE COMPRESIÓN NORMALIZADA (NCD) 7
Complejidad de Largo de Programa *7*
Distancia de Similitud Universal *8*
Distancia de Compresión Normalizada *10*
ANÁLISIS Y COMPARACIÓN DE SECUENCIAS BIOLÓGICAS 11
LA NCD APLICADA AL ANÁLISIS Y COMPARACIÓN DE SECUENCIAS 12

INTRODUCCIÓN A LA FILOGENIA **14**

Análisis filogenético *15*
Métodos basados en distancia *16*
Método de Máxima Parsimonia *18*
Método de Máxima Verosimilitud *19*

ANÁLISIS DE COMPRESORES **21**

COMPRESOR NORMAL 21
ALGORITMOS DE COMPRESIÓN DE DATOS *22*
LZ77 ..... *22*
LZ78 ..... *23*
Ordenamiento de bloques de Burrows – Wheeler *23*
IMPLEMENTACIONES EVALUADAS 25
GZIP *25*
BZIP2 ..... *25*
LRZIP *26*
GenCompress *27*
Otros Compresores *27*
RESULTADOS EXPERIMENTALES 28
Análisis de la capacidad de compresión *29*
Análisis de tiempo de ejecución *30*
Análisis de la propiedad de idempotencia *31*
Análisis de la propiedad de simetría *32*

NCD APLICADA A LA FILOGENIA **34**

INTRODUCCIÓN A LA MÉTRICA AGREEMENT 34
PAUP 36
INFERENCIA DE ÁRBOLES FILOGENÉTICOS 38
Datasets *39*
Preparación de los datos *39*
Procesamiento en PAUP *41*
Matriz de distancia NCD *42*
Utilización de la NCD en PAUP *43*
Validación de la métrica NCD *44*
Comparación *46*

ANÁLISIS DE LOS RESULTADOS OBTENIDOS **48**

COMPARACIÓN GENERAL ENTRE LA NCD Y LAS OTRAS MÉTRICAS DE PAUP 49
COMPARACIÓN DE ÁRBOLES IDÉNTICOS 53
FUNCIÓN DE SIMILITUD Y CANTIDAD DE SECUENCIAS DEL DATASET 54

CONCLUSIONES **57**

TRABAJOS FUTUROS **59**

ANEXO I **61**
REFERENCIAS **68**

Abstract

La similitud de secuencias es la noción matemática primordial para los estudios biológicos sobre filogenética. Los enfoques clásicos aportados desde la bioinformática para la resolución de este problema utilizan algoritmos de alineamiento para obtener una noción de similitud entre dos secuencias biológicas. Sin embargo estos tipos de métodos no son los adecuados cuando las secuencias a comparar son muy diferentes entre sí [13]. Por este motivo, las medidas de similitud entre secuencias que no requieren alineamiento son una nueva alternativa para abordarlos. En [1] Paul Vitányi definió la Medida de Similitud Universal (Universal Similarity Metric) basada en la complejidad de largo de programa definida por Chaitin-Kolmogorov [3][4]. Dado que esta noción no es computable, el mismo Vitányi propuso su aproximación utilizando compresores estándares de texto: Distancia de Compresión Normalizada (Normalize Compression Distance) [2].

En este trabajo abordamos el estudio del uso de la NCD definida por Vitányi a un problema importante de la biología como lo es la inferencia de árboles filogenéticos. El mismo Vitányi abordó ligeramente este tema, sin profundizar ni realizar un análisis crítico al respecto [1][5].

En primer lugar, estudiamos este problema y las herramientas informáticas existentes que se utilizan para resolverlo. Luego analizamos y comparamos el uso de distintos compresores que pueden ser utilizados para aplicar el método de Vitányi, analizando sus limitaciones y ventajas. Para la realización de los experimentos se utilizaron datasets tradicionales, utilizados en distintas publicaciones relacionadas con este tema y que conforman la bibliografía de este trabajo; también se desarrollaron adaptadores que permiten utilizar la métrica basada en la NCD dentro del PAUP. El PAUP es un paquete de software utilizado para inferir árboles evolutivos, ampliamente difundido en la comunidad bioinformática [28]. Incorporar la NCD al conjunto de métricas disponibles dentro de PAUP, nos permitió comparar nuestros resultados, directamente con los obtenidos mediante los métodos actualmente existentes.

Introducción

El análisis de secuencias biológicas es una de las actividades principales de la bioinformática. Esta disciplina se apoya en el paradigma de que una secuencia determina la función molecular y finalmente la función de las moléculas en la célula. Por ende, las diferencias y las similitudes de las secuencias son analizadas con el objetivo de poder inferir las relaciones estructurales, funcionales y evolutivas.

Los enfoques clásicos aportados por la bioinformática para buscar similitudes entre secuencias se basan en algoritmos de alineamiento. Generalmente estos métodos tienden a ignorar los procesos de recombinación genética. Esta limitación derivó en el desarrollo de métodos para buscar similitudes entre secuencias que no utilicen algoritmos de alineamiento.

Esta tesis estudia el problema del análisis de secuencias biológicas utilizando un método libre de alineamiento. Este método se basa en la distancia de compresión normalizada (NCD) propuesta por Vitányi [5] para establecer una noción de similitud entre secuencias.

Este capítulo describe las motivaciones de este trabajo y repasa la información necesaria para entender los conceptos básicos relacionados con el análisis de secuencias biológicas y la distancia de compresión normalizada. De esta manera brindamos el marco necesario para poder justificar la utilización de esta distancia cómo parte de un método libre de alineamiento, para el análisis de secuencias biológicas.

La primera sección de este capítulo brinda una breve introducción sobre la biológica molecular, las secuencias biológicas y su importancia en los procesos de los seres vivos. Luego se presenta una introducción sobre las técnicas predominantes utilizadas para el análisis de secuencias biológicas. Finalmente se explican los fundamentos de la distancia de compresión normalizada y se analiza su utilización en un problema reconocido de la biología molecular: la inferencia de árboles filogenéticos.

Conceptos básicos de biología molecular

A continuación se abordarán los conceptos básicos sobre biología molecular necesarios para el desarrollo y la compresión de este trabajo.

Secuencias biológicas

El *ácido desoxirribonucleico* (ADN) es un ácido nucleído que contiene la información genética usada en el desarrollo y el funcionamiento de la mayoría de los seres vivos. La función principal de las moléculas de ADN es la de ser portador y transmisor de la información genética entre las sucesivas generaciones de organismos vivos.

Desde el punto de vista químico, el ADN está compuesto de nucleótidos o bases: la adenina, la timina, la citosina y la guanina. Estas bases son generalmente referenciadas por sus iniciales: A, T, C y G. Estructuralmente, el ADN está formado por dos *hebras* (strands) o cadenas de bases entrelazadas que conforman una doble hélice de tal manera que las bases de tipo A se aparean solo con las bases T y las bases C lo hacen solamente con las bases G. Debido a esta propiedad de apareamiento, una hebra de la molécula puede ser determinada unívocamente examinando la otra hebra y viceversa.

Una cadena de ADN presenta dos tipos de regiones: ADN codificante y ADN no codificante. El ADN codificante está constituido por los genes que son las unidades de información hereditaria. El ADN no codificante no codifica proteínas, razón por la cual también se lo denomina ADN basura. Sin embargo, en los últimos años se ha descubierto que ciertas regiones cumplen importantes funciones biológicas.

Las *proteínas* son macromoléculas codificadas por los genes que llevan a cabo la mayoría de las actividades biológicas de las células. Desde el punto de vista químico están formadas por cadenas lineales de aminoácidos.

Aminoácido	Abreviación	Símbolo
Alanina	Ala	A
Arginina	Arg	R
Asparagina	Asn	N
Ácido aspártico	Asp	D
Cisteina	Cys	C
Glutamina	Gln	Q
Ácido Glutámico	Glu	E
Glicina	Gly	G
Histidina	His	H
Isoleucina	Ile	I
Leucina	Leu	L
Lisina	Lys	K
Metionina	Met	M

Fenilalalina	Phe	F
Prolina	Pro	P
Serina	Ser	S
Treonina	Thr	T
Triptófano	Trp	W
Tirosina	Tyr	Y
Valina	Val	V

Tabla 1 - Lista de aminoácidos

Las proteínas son sintetizadas por las regiones codificantes del ADN (genes). Las regiones codificantes son caracterizadas por tripletes de bases (tres bases contiguas), por ende hay 64 (4^3) tripletas posibles. Estas tripletas reciben el nombre de *codones*. En el proceso de "traducción" de ADN a proteínas, cada codón se mapea con un único aminoácido. Este mapeo recibe el nombre de *código genético*. Dado que hay solo 20 *aminoácidos* (Tabla 1) y 64 *codones*, hay aminoácidos que son mapeados por más de un codón.

Una molécula de ADN y sus proteínas constituyen un *cromosoma* y todo el ADN que constituye los cromosomas de un organismo recibe el nombre de *genoma*.

Fundamentos de la Distancia de Compresión Normalizada (NCD)

En esta sección se describirá la base teórica en la que se basa la noción de similitud aplicada en este trabajo para la clasificación de cadenas de naturaleza biológica. Para ello es necesario, en primer término, presentar una descripción de la teoría de complejidad de largo de programa. Este concepto fue utilizado por Vitányi para obtener la definición de la métrica de similitud universal, que le permitió definir formalmente una noción de distancia entre cadenas. La complejidad de largo de programa es una noción no computable. Por consiguiente, la noción de similitud universal también es no computable. Sin embargo, Vitányi definió una aproximación denominada *distancia de compresión normalizada*, basada en la utilización de compresores standards.

Complejidad de Largo de Programa

La complejidad de largo de programa [1][3] puede verse como una cuantificación absoluta y objetiva de la cantidad de información que hay en una cadena.

Dado $x \in \{0,1\}^*$, la complejidad de largo de programa está definida por la función H, donde H(x) es la longitud del programa más corto que computa x en una máquina universal de Turing.

Se denota como x* a un programa de tamaño H(x) que calcula x.

La complejidad condicional H(x|y) se define como la longitud del programa más corto que computa x, si un programa de longitud mínima que computa y es dado como input. Intuitivamente H(x|y) representa cuánto cuesta generar x teniendo, sin costo alguno, el mejor programa (el más corto) que genera y.

La complejidad conjunta H(x,y) es igual al tamaño del mínimo programa que computa (x,y), donde la coma representa una función inyectiva que dadas dos cadenas devuelve una sola cadena.

En [1] Chaitin demostró la siguiente ecuación:

$$(1) \qquad H(x,y) \approx H(x|y) + (y)$$

Despejando, puede verse que:

$$(2) \qquad H(x|y) \approx H(x,y) - H(y)$$

Entonces, H(x|y) puede interpretarse como cuánto cuesta generar x e y menos lo que cuesta generar y.

La complejidad de largo de programa es un concepto no computable y sólo puede ser aproximado superiormente.

Distancia de Similitud Universal

Dadas dos cadenas x e y, la distancia de *similitud universal* presentada por Vitányi en [5] se define como:

$$(3) \qquad d(x,y) = \frac{max\{H(x|y),H(y|x)\}}{max\{H(x),H(y)\}}$$

Existe una interpretación natural de d(x,y): si H(y) ≥ H(x), puede rescribirse la definición anterior como:

$$d(x,y) = \frac{H(y)-I(x:y)}{H(y)}$$

En esta definición, I(x:y) representa la información mutua presente entre x e y. Cabe destacar que I cumple con la propiedad de simetría, es decir: I(x:y) = I(y:x). Luego, la distancia d(x, y) entre x e y es el número de bits de información que no son compartidos entre las dos cadenas, por bit de información que podrían ser maximalmente compartidos por ambos.

Otra manera de interpretar esta noción, es basándose en la ecuación (2) y sabiendo que:

$$(4) \quad H(x,y) \approx H(y,x)$$

Sabemos lo siguiente por (2):

$$H(x|y) \approx H(x,y) - H(y)$$
$$H(y|x) \approx H(y,x) - H(x)$$

Luego por (4) el numerador de (3) puede ser rescrito como:

$$max\{H(x|y), H(y|x)\} = max\{H(x,y) - H(y), H(y,x) - H(x)\} \approx H(x,y) - min\{H(x), H(y)\}$$

Reemplazando en (3):

$$(5) \quad d(x,y) = \frac{H(x,y) - \min\{H(x),H(y)\}}{\max\{H(x),H(y)\}}$$

Suponiendo H(x) > H(y), nos queda:

$$(6) \quad d(x,y) = \frac{H(x,y) - H(y)}{H(x)}$$

En [4] Chaitín demostró que, si x e y son muy distintos se cumple que:

$$H(x,y) \approx H(x) + H(y)$$

Entonces si suponemos que x e y son muy distintos, tras reemplazar en (6) nos queda:

$d(x,y) = \frac{H(x)+H(y)-H(y)}{H(x)}$ (máximo valor posible)

Luego se respeta la idea intuitiva de que si x e y son dos cadenas muy distintas, la distancia entre ellas es la máxima posible.

Analicemos el caso contrario: supongamos que x e y son la misma cadena.

Sabemos que:

$$x = y \rightarrow H(x) = H(y)$$

En [1] Chaitín demostró que:

$$H(x,x) \approx H(x)$$

Luego, reemplazando en (6) obtenemos:

$$d(x,y) = \frac{H(x) - H(x)}{H(x)}$$

Nuevamente se respeta la idea intuitiva. La distancia de una cadena a sí misma es cero.

Distancia de Compresión Normalizada

En [5], Vitányi demuestra que la distancia de similitud universal es una métrica. Incluso se muestra que es universal en el sentido que cualquier otra métrica que expresa alguna similitud entre dos objetos puede ser minimizada (comprimida) por d(x,y). Esto quiere decir que si dos objetos son similares según una métrica, también serán similares en el sentido de la métrica de similitud universal. Sin embargo, como está basada en la noción no computable de complejidad de largo de programa de Kolmogorov/Chaitín, la distancia de similitud universal tampoco es computable. Para poder aplicar el concepto teórico, Vitányi realizó una aproximación basada en compresores estándares [2]. La aproximación resultante de la distancia de similitud universal (3) se denomina distancia de compresión normalizada (NCD) y está definida como:

$$NCD(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}$$

En esta definición C representa la aproximación de H utilizando un compresor estándar, C(x) el tamaño de la compresión de 'x', y 'xy' la concatenación de 'x' con 'y'.

La NCD es un número no negativo, menor o igual a 1 + ε, que representa qué tan diferentes son dos cadenas entre sí. Valores de NCD cercanos a 1 indican menor similitud mientras que valores cercanos a 0 indican mayor similitud. El ε de la cota superior se debe a imperfecciones en las técnicas de compresión, pero para algoritmos de compresión estándares es típicamente inferior a 0,1.

La teoría desarrollada por Vitányi en [5] para la complejidad algorítmica no es extensible directamente para la aproximación NCD. Luego, en [2], Vitányi desarrolló

la teoría de NCD basada en el concepto de compresor normal y se demostró que la NCD es una métrica de similitud quasi-universal con respecto a un compresor normal C.

Si bien una mejor compresión siempre aproximará mejor a la complejidad H, no sucede necesariamente lo mismo con la noción de distancia de similitud universal. Naturalmente esto se debe a que su definición involucra el cálculo de un cociente entre dos estimaciones, y la estimación de la complejidad condicional mediante una resta.

Análisis y comparación de secuencias biológicas

La comparación de secuencias biológicas es la base de la mayoría de las aplicaciones bioinformáticas. Su objetivo es poder inferir relaciones estructurales, funcionales y evolutivas de los organismos que contienen las secuencias.

Durante la duplicación de ADN se pueden generar cambios en las bases del ADN resultante. Frecuentemente estos cambios generan modificaciones en la estructura de las proteínas que son codificadas a partir de este ADN, alterando así su función. A este proceso se lo denomina mutación. Las mutaciones pueden ocurrir por diversas razones desde errores en la duplicación hasta la exposición a factores ambientales como virus, agentes químicos o radiación. Existen distintos tipos de mutaciones, a la más simple se la conoce como mutación simple (*single point mutation*), ésta consiste en la sustitución de una base por otra. Otras posibles alteraciones son las inserciones o eliminaciones, estas ocurren cuando una o más bases son agregadas o bien eliminadas del ADN original. Las mutaciones van pasando de generación en generación, por lo tanto cada descendiente diferirá de la secuencia original a distintos niveles. Si dos secuencias comparten el mismo ancestro se las denomina *homólogas*. Estas variaciones en el ADN juegan un papel esencial en el proceso evolutivo y la selección natural.

Los métodos más utilizados para comparar secuencias biológicas son los que se basan en el a*lineamiento* de cadenas de caracteres. Este proceso consiste en posicionar una cadena sobre la otra y remarcar los símbolos que presentan en común, representándolos con líneas verticales (|). Además de detectar los símbolos que se comparten entre las secuencias, el alineamiento permite detectar diferencias entre las mismas. Estas diferencias podrían ser por ejemplo la presencia de un símbolo en una de las secuencias que no existe en la otra, lo que corresponde a una mutación simple. Este proceso también reconoce la inserción o eliminación de uno o

más símbolos entre las secuencias. A estos conjuntos de caracteres (insertados o eliminados) se los denomina *gaps* y se los representan con el símbolo "-". En teoría, el alineamiento permite examinar mutaciones producidas entre distintas generaciones de ADN. La idea fundamental del alineamiento aplicado a secuencias biológicas es la detección de relaciones de homología entre las mismas.

Se han desarrollado diversos algoritmos de alineamiento de secuencias biológicas. Entre ellos se destacan el algoritmo de alineamiento global desarrollado por Needleman y Wunsch [25] y el algoritmo de alineamiento local desarrollado por Smith y Waterman [26]. Estos algoritmos son mejorados por medio del uso de matrices de substitución y de penalidades por *gaps*. Las matrices de substitución son utilizadas para asignar un valor al alineamiento de cada par de símbolos del alfabeto. De esta manera se establece un valor diferente para cada substitución de símbolos. Por ejemplo la substitución de A por G podría penalizar el alineamiento en mayor medida que la substitución de A por C o viceversa. Por otro lado los valores de penalidades por *gaps* establecen un valor de penalidad a una serie de inserciones o eliminaciones de n símbolos contiguos. El valor resultante del alineamiento se obtiene de la suma de los valores de cada penalidad. El uso de las matrices de substitución y penalidades por *gaps* hace posible aplicar el algoritmo en el estudio de secuencias biológicas utilizando distintos modelos de evolución.

La NCD aplicada al análisis y comparación de secuencias

La distancia de compresión normalizada es una métrica definida para poder comparar cadenas de caracteres en general. Esta característica permite utilizarla en diferentes dominios. Por ejemplo, en [24] se comprobó que esta distancia puede ser utilizada para clasificar automáticamente documentos de distinta naturaleza (literatura, código fuente en distintos lenguajes de programación, etc.).

En particular, en este trabajo aplicaremos esta distancia a un problema específico de biología. Dentro de los problemas de biología susceptibles a ser resueltos mediante métodos automáticos, se encuentran el problema de buscar, dado un fragmento de secuencia biológica, las secuencias similares en bases de datos de gran tamaño; la inferencia de relaciones evolutivas de un conjunto de secuencias (filogenia); problemas de clasificación y agrupamiento (clustering) de secuencias. El factor común de estos problemas es la necesidad de comparar secuencias biológicas de alguna manera.

Como hemos explicado anteriormente, tanto el ADN como las proteínas son usualmente representados por cadenas de caracteres. Desde este punto de vista, la única diferencia entre ambas secuencias es el alfabeto utilizado. En el caso del ADN se utiliza un alfabeto de cuatro caracteres donde cada carácter representa una base o nucleótido mientras que las secuencias de proteínas utilizan un alfabeto de veinte caracteres donde cada uno de ellos representa a un aminoácido.

ADN	A T C G
Proteínas	A R N D C Q E H I L K M F P S T W Y V

Tabla 2 – Alfabetos utilizados para representar proteínas y ADN

Justamente el hecho de representar al ADN o a las proteínas como secuencias de caracteres permite que podamos utilizar la distancia NCD desarrollada por Vitányi como una alternativa a los métodos tradicionales para la comparación de secuencias biológicas.

Desde el punto de vista intuitivo, la NCD puede ser aplicada para inferir relaciones evolutivas ya que si dos organismos tienen un ancestro en común, seguramente sus secuencias tienen varias regiones de ADN en común. Por lo tanto, al utilizar la NCD para medir la similitud entre las secuencias de estos organismos, el compresor, al comprimir la concatenación de las cadenas, detectará las regiones comunes a ambas, lo que permitirá inferir la similitud de ambas secuencias. Si las secuencias son totalmente diferentes entre sí, el compresor no detectará redundancia, y por lo tanto la tasa de compresión será muy baja, indicando que dichas secuencias no son similares.

De manera similar, la NCD puede ser utilizada para agrupar y/o clasificar un conjunto secuencias biológicas. Para realizar esto, en primer lugar se debería calcular la distancia entre todos los pares de secuencias obteniendo de esta manera una matriz de distancias. Luego, se procedería a aplicar algún algoritmo de clustering a esta matriz de distancia, obteniendo finalmente una partición del conjunto de secuencias inicial.

Introducción a la filogenia

El análisis filogenético de una familia de secuencias de ADN consiste en determinar cómo se ha formado la familia durante su proceso evolutivo. Las relaciones evolutivas son representadas a través de un árbol filogenético cuyos nodos representan a las secuencias y sus ejes a la relación evolutiva entre ellas.

El objetivo de la filogenia es la construcción de árboles filogenéticos a partir de la inferencia de las relaciones evolutivas existentes entre las secuencias a estudiar.

Los métodos utilizados en el análisis filogenético están sumamente relacionados con los empleados en el alineamiento de secuencias. Tal como dos secuencias muy similares entre sí pueden ser fácilmente alineadas, un grupo de secuencias similares pueden ser fácilmente organizadas en un árbol. En cambio cuando las secuencias presentan una gran cantidad de cambios evolutivos entre ellas, son más difíciles de alinear. Lo mismo sucede con el análisis filogenético; cuando las secuencias son muy diferentes entre sí, se incrementa la cantidad de posibles árboles filogenéticos distintos que pueden representar sus variaciones.

Los métodos utilizados para inferir árboles filogenéticos pueden ser clasificados en dos grandes grupos: métodos basados en distancia y métodos basados en caracteres. Los primeros resumen la información de un alineamiento múltiple en una matriz de distancias entre las secuencias. Generalmente, primero utilizan el alineamiento para calcular la distancia (de acuerdo a un modelo de evolución dado) entre las secuencias y luego tratan de reconstruir el árbol evolutivo a partir de dichas distancias. Por su parte, los métodos basados en caracteres hacen uso de la información de cada región del alineamiento múltiple para inferir la mejor hipótesis. Los métodos de máxima parsimonia y máxima verosimilitud están dentro de este último grupo.

Cada uno de estos métodos utiliza un tipo de análisis diferente y resuelve mejor ciertos tipos de secuencias, en cada caso con sus ventajas y desventajas. En la actualidad existen programas que cuentan con una implementación para cada uno de estos métodos. Entre ellos podemos mencionar al PHYLIP [27], PAUP [28], TNT [29] y BLAST [30].

Análisis filogenético

Un *árbol evolutivo* o *filogenético*[1] es un grafo de dos dimensiones que muestra relaciones evolutivas entre organismos. El árbol está compuesto por nodos externos u hojas que representan las secuencias y nodos internos que representan las relaciones filogenéticas establecidas entre ellas.

Existen dos tipos de árboles evolutivos: *jerárquicos* y *no jerárquicos*. Los *árboles jerárquicos* (rooted trees) son aquellos en los cuales todos los nodos del árbol comparten un ancestro en común que está representado por la raíz del árbol. En cambio los *árboles no jerárquicos* (unrooted trees) no cuentan con una jerarquía definida.

En la Figura 1 se muestra un árbol jerárquico compuesto de cuatro secuencias: A, B, C y D. Las secuencias A y B son derivadas de un ancestro en común representado por el nodo A/B que es el padre de ambas. Las secuencias C y D están relacionadas de la misma manera por el nodo C/D. El ancestro A/B y C/D tienen un ancestro en común que está representado por el nodo raíz del árbol. Es importante observar que cada nodo en el árbol representa una división en el camino evolutivo de los genes de dos especies distintas. Cualquier cambio evolutivo en una rama será independiente de cualquier otro cambio en otra rama.

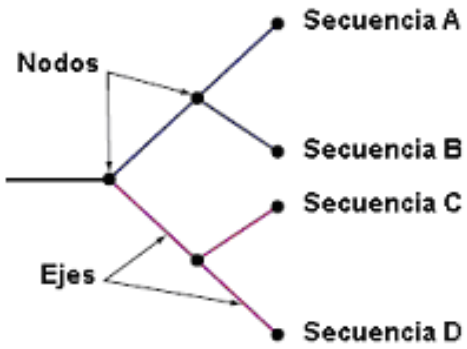


Figura 1 – Árbol filogenético jerárquico

Una representación alternativa es el uso de árboles no jerárquicos. En la Figura 2 se muestran las relaciones evolutivas entre las secuencias A, B, C y D utilizando un árbol no jerárquico.

[1] Árbol evolutivo, de evolución o filogenético se refieren a lo mismo y serán utilizados de manera indistinta.

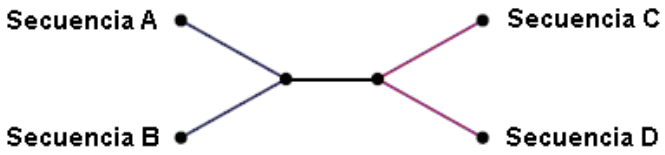


Figura 2 - Árbol filogenético no jerárquico

Los árboles no jerárquicos también muestran las relaciones evolutivas entre secuencias, pero esta representación no muestra en donde se encuentra el ancestro común a todas las secuencias. Este árbol puede ser fácilmente convertido a árbol jerárquico tan solo agregándole la raíz. La incógnita es entre que pares de nodos debe insertarse el ancestro común.

El número de posibles árboles jerárquicos se incrementa rápidamente con el número de secuencias biológicas a analizar. En la Tabla 3 se muestra el número de posibles árboles evolutivos en función del número de secuencias.

Cantidad de secuencias	Cantidad árboles jerárquicos	Cantidad árboles no jerárquicos
3	3	1
4	15	3
5	105	15
-	-	-
7	10395	954

Tabla 3 – Cantidad de árboles en función de la cantidad de secuencias

Métodos basados en distancia

Los métodos de filogenia basados en distancias utilizan la información del alineamiento múltiple para calcular las distancias entre las secuencias. En otras palabras, estos algoritmos transforman la información del alineamiento en una matriz de distancias. Posteriormente reconstruyen el árbol filogenético con la información de dicha matriz.

Generalmente estos métodos se dividen en dos pasos. El primero consiste en construir una matriz de distancias utilizando la información obtenida en el alineamiento múltiple, y el segundo paso consiste en aplicar un método de clustering a la matriz de distancia para construir un árbol filogenético.

En el alineamiento múltiple de secuencias se calcula un puntaje de similitud entre cada par de secuencias. Este puntaje determina la distancia entre las

secuencias de dicho par. Dicho puntaje es calculado aplicando un modelo en particular.

Una vez obtenida la matriz de distancias se utiliza un método de clustering para construir el árbol filogenético. Los métodos de clustering más utilizados para generar árboles filogenéticos son el Neighbor-Joining y UPGMA.

Método de clustering Neighbor-Joining

El método de Neighbor-Joining (NJ) [18] es un algoritmo recursivo de clustering utilizado para la construcción de árboles filogenéticos. Éste toma como entrada una matriz de distancias y devuelve un árbol filogenético. Este algoritmo es especialmente conveniente cuando el índice de evolución entre las secuencias es variable.

Por ser una heurística, este algoritmo no asegura que el árbol construido sea el óptimo, pero entrega muy buenos resultados en varios modelos evolutivos. Su principal ventaja es la eficiencia computacional dado que su orden de complejidad es polinomial. Por lo tanto puede ser utilizado para analizar grandes conjuntos de datos. El árbol generado es *no jerárquico*, aunque puede ser jerarquizado utilizando un método que pueda determinar la raíz del mismo.

Método de clustering UPGMA

Al igual que el algoritmo NJ, el método de clustering UPGMA [20] es un algoritmo recursivo de clustering utilizado para la construcción de árboles filogenéticos. Éste toma como entrada una matriz de distancias y devuelve un árbol filogenético.

Inicialmente cada secuencia de la matriz forma su propio cluster. De forma recursiva, en cada paso se unen los dos clusters mas cercanos y se recalculan las distancias entre los clusters resultantes. La distancia entre clusters se define como la media de las distancias que hay entre los miembros de cada cluster.

UPGMA asume una tasa de evolución constante (hipótesis del reloj molecular). Por este motivo no se recomienda su uso para inferir árboles filogenéticos cuando el conjunto de secuencias no cumpla con esta hipótesis.

Método de Máxima Parsimonia

El método de Máxima Parsimonia se basa en el principio que determina que el árbol filogenético que represente la evolución mínima será el que mejor estime a la filogenia real. El objetivo del mismo es predecir él o los árboles filogenéticos que minimicen el número de pasos requeridos para generar la evolución mínima observada en las secuencias.

El método requiere un alineamiento múltiple del conjunto de secuencias. El resultado de este alineamiento mostrará las posiciones similares de las secuencias por medio de columnas verticales. Para cada columna, se identifican los árboles filogenéticos que requieran el menor número de cambios evolutivos para formar los cambios observados. Finalmente son seleccionados los árboles que requieren el menor número de cambios para todas las posiciones (columnas del alineamiento) de las secuencias.

El algoritmo cuenta con los siguientes pasos:

```
M = Alineamiento múltiple del conjunto de secuencias

A' = Generar todos los árboles No Jerárquicos del conjunto de secuencias

D = Mapa: árbol -> Distancia

Para cada columna c en M

       Para cada árbol a en A'

              D[a] = D[a] + Cantidad de cambios evolutivos (c , a)

       Fin Para

Fin Para

Resultado = { a / D[a] = Min { D[a'] / a' Pertenece A} }
```

El algoritmo no es complicado y garantiza que encontrará el árbol que describa la mínima evolución de las secuencias ya que todos los posibles árboles son examinados. Por esta razón este método es lento y generalmente no es utilizado para grandes conjuntos de secuencias ni tampoco cuando existe mucha variación entre las mismas.

Este es un método exhaustivo, por lo que los paquetes de software que lo implementan, suelen contar con heurísticas para analizar sólo un conjunto de árboles representativos, en lugar de recorrer todos los posibles árboles.

Método de Máxima Verosimilitud

El método de Máxima Verosimilitud (o simplemente máxima similitud) utiliza cálculos probabilísticos con el objetivo de encontrar el árbol filogenético que mejor represente las variaciones de las secuencias.

El algoritmo consiste de los siguientes pasos:

1. Se calculan todos los posibles árboles filogenéticos *no jerárquicos* que se pueden generar con las secuencias biológicas.

2. Para cada árbol filogenético se calcula la probabilidad de que dicho árbol haya sido generado por el conjunto de secuencias biológicas.

3. Por último el árbol filogenético elegido será aquel que tenga mayor probabilidad.

El método de máxima verosimilitud es similar al de parsimonia en el sentido de que todos los posibles árboles filogenéticos son recorridos durante el análisis. Otra característica que comparten es que el árbol elegido será aquel que tenga la menor cantidad de cambios. Sin embargo, este método puede ser utilizado para investigar relaciones entre secuencias más diversas y en distintas condiciones que no son manejadas muy bien por el método de máxima parsimonia.

Una ventaja de este método es que a diferencia del resto puede utilizar una vasta información estadística acercar de la tasa de evolución que presenta el conjunto de secuencias biológicas. Además cuenta con otro tipo de información, por ejemplo la tasa de evolución puede variar entre especies, entre genes o a través del tiempo. Por último, este análisis calcula la probabilidad exacta para cada árbol filogenético no jerárquico. Con esta información se pueden realizar comparaciones cuantitativas entre distintos árboles filogenéticos, por ejemplo se puede decir que un árbol es exactamente un por ciento más probable que otro. Este tipo de comparaciones cuantitativas no son fáciles de calcular con otros tipos de métodos como el de máxima parsimonia.

La principal desventaja de este método es que es computacionalmente intenso; aún más que el método de máxima parsimonia, ya que no sólo recorre todos los posibles árboles filogenéticos sino también debe realizar cálculos de

probabilidad para cada filogenia analizada. Por lo tanto este método es factible únicamente para conjuntos de secuencias pequeños.

Análisis de compresores

Como explicamos anteriormente, la métrica de similitud universal está basada en la medida de complejidad de largo de programa definida por Chaitín en [4]. Esta medida de complejidad no es computable. Por ende, para poder aplicar este concepto teórico, es necesario aproximarlo por medio de algún procedimiento computable. La idea central del trabajo de Vitányi [5] es aproximar esta complejidad por medio de compresores de datos. De allí nace la definición de la distancia de compresión normalizada.

La elección del compresor utilizado para aproximar la complejidad de largo de programa es un paso de vital importancia en la aplicación de la NCD. Por este motivo es necesario estudiar las características que tienen que tener los compresores para poder ser utilizados como aproximación de esta complejidad. En [1], Vitányi define 4 propiedades que deben cumplir los compresores para ser utilizados en el cálculo de la NCD. Los compresores que cumplen con estas propiedades se los denomina *Compresores Normales*.

Compresor Normal

Si bien Vitányi no define un criterio para la elección de un compresor que sirva para aproximar la complejidad de largo de programa, en [1] define la noción de *Compresor Normal* de la siguiente manera: Un compresor C es *normal* si satisface las siguientes ecuaciones:

a) Idempotencia

$$C(xx) \approx C(x)$$

b) Monotonicidad

$$C(xy) \geq C(x)$$

c) Simetría

$$C(xy) \approx C(yx)$$

d) Distributividad

$$C(xy) + C(z) \leq C(xz) + C(yz)$$

Estas propiedades son cumplidas por la función H. Por lo tanto es lógico pedir que el compresor elegido para aproximar H también las cumpla de manera aproximada.

Desde el punto de vista de la teoría de complejidad algorítmica, la propiedad a) representa que el costo de calcular xx es similar al costo de calcular x. En el plano de los compresores, es esperable que un compresor aproveche las repeticiones exactas dentro del archivo de entrada. En cuanto a la propiedad b) es trivial darse cuenta que todos los compresores la cumplen. En el caso de las propiedades c) y d) no es simple determinar sí son satisfechas por cualquier compresor.

En las publicaciones que utilizan la NCD como medida de similitud entre secuencias de ADN o proteínas, no suelen corroborar si los compresores utilizados son normales. Este es el caso de los siguientes trabajos: [31], [32], [11] y [10].

Algoritmos de compresión de datos

Los algoritmos de compresión sin pérdida de información se pueden clasificar en dos tipos de acuerdo a la manera en la que procesan el archivo de entrada. Por un lado se encuentran los algoritmos *secuenciales* ya que procesan y codifican el archivo de entrada de manera secuencial hasta detectar el fin de archivo, mientras que por el otro se encuentran los algoritmos de *bloques* pues dividen el archivo de entrada en bloques y luego procesan y codifican cada bloque de manera individual.

Esta forma de clasificar a los compresores adquiere relevancia porque, a priori, se puede establecer relaciones entre la forma de procesar su entrada y las propiedades que definen a los compresores normales. Por ejemplo los compresores secuenciales que utilicen ventanas pequeñas difícilmente cumplan con la propiedad de simetría con archivos de gran tamaño. En cambio los compresores que procesan por bloques pueden tener mejores resultados con respecto a la simetría si todo el archivo a comprimir cabe en un solo bloque. Por este motivo ahora daremos una breve descripción de los algoritmos de compresión utilizados en este trabajo, teniendo en cuenta esta perspectiva.

LZ77

Es un algoritmo secuencial [8] que se basa en encontrar subsecuencias repetidas dentro de la secuencia a comprimir. Introduce el término de "ventana deslizante", mediante la cual, dada una posición de la secuencia, hay un registro de cuáles caracteres aparecieron antes. El tamaño de las subsecuencias se encuentra acotado por un *lookahead* que determina cuántos bytes "hacia delante" es capaz de inspeccionar el algoritmo.

Una ventana de 32K significa que el compresor tiene un registro de cuáles fueron los últimos 32768 (32·1024) caracteres. Cuando la próxima secuencia de caracteres a compactar es idéntica a una que puede ser encontrada por la ventana deslizante, se reemplaza por dos números: una distancia, representando cuanto hay que retroceder en la ventana para encontrar el inicio de la secuencia, y una longitud, que determina la cantidad de caracteres que ambas secuencias poseen iguales.

Este algoritmo implícitamente asume que los patrones en el archivo de entrada se encuentran cercanos entre si. Por ende, podemos concluir que las propiedades de normalidad van a estar directamente afectadas por el tamaño de la ventana deslizante que utiliza la implementación de este algoritmo.

LZ78

El algoritmo también es secuencial, pero a diferencia del algoritmo LZ77 no utiliza el concepto de ventana deslizante. En su lugar, utiliza un diccionario que contiene las cadenas previamente encontradas. El diccionario está compuesto por un índice y una palabra asociada a dicho índice. Al principio el diccionario está vacío, y su tamaño está limitado por la cantidad de memoria disponible.

El algoritmo LZ78 [9] codifica la información por medio de pares (índice, carácter siguiente). El índice es un puntero a la entrada del diccionario que contiene el emparejamiento (*matching*) más largo que se puede hallar a partir de la posición actual; el carácter siguiente indica el símbolo que precede en el archivo de entrada al emparejamiento. Un índice de 0 indicará que la palabra no se encuentra en el diccionario, entonces 'carácter siguiente' es la información aportada por el par.

Ordenamiento de bloques de Burrows – Wheeler

El algoritmo de ordenamiento de bloques de Burrows – Wheeler [10] es un algoritmo de compresión de datos. La idea del algoritmo no es procesar el archivo de entrada de manera secuencial, sino la de procesar bloques de texto del mismo.

El algoritmo cuenta con dos etapas. En la primera etapa se aplica una transformación reversible a un bloque de texto para obtener una permutación de dicho bloque, que permite una compresión por medio de algoritmos sencillos. La transformación tiende a agrupar los caracteres de tal manera que la probabilidad de encontrar caracteres idénticos cercanos entre sí, se incrementa notablemente. En esta etapa se transforma una cadena S de N caracteres, formando las N rotaciones

de S, ordenándolas lexicográficamente y extrayendo el último caracter de cada una de dichas rotaciones. Una nueva cadena L se forma con dichos caracteres, donde el i-ésimo caracter de L es el último caracter de la i-ésima rotación ordenada de S. El punto clave del algoritmo es que, dado L y el índice I correspondiente a la posición de la cadena S en la lista ordenada de las rotaciones, existe un algoritmo eficiente para computar S. Por lo tanto solo basta codificar la cadena L y el índice I.

Durante el proceso de ordenamiento lexicográfico de las rotaciones, aquellas rotaciones cuyos caracteres iniciales son idénticos, quedan contiguas. Como los caracteres iniciales son adyacentes a los caracteres finales, los caracteres consecutivos en L son adyacentes a cadenas similares en S. Por lo tanto, L podrá ser fácilmente comprimida por simples algoritmos. Justamente la segunda etapa del algoritmo se encarga de comprimir la cadena L mediante la aplicación de algoritmos localmente adaptativos como move–to–front [10], en combinación con otros tipos de codificación (Huffman [33] o aritmética).

Implementaciones evaluadas

GZIP

Gzip es un compresor de archivos basado en una variación del algoritmo LZ77 [8]. El mecanismo de compresión posee un overhead constante de algunos bytes correspondiente al encabezado del archivo y 5 bytes por cada bloque de 32 KB.

El algoritmo se basa en encontrar secuencias repetidas en la cadena de entrada, reemplazando las ocurrencias repetidas por un puntero a la aparición previa en la forma de un par *(distancia, longitud)*. Para ello se utiliza el concepto de *"ventana deslizante"*: dada una posición de la secuencia, hay un registro de cuáles fueron los caracteres anteriores. En el caso particular del Gzip, el tamaño de la ventana (y por lo tanto de la distancia máxima) es de 32 KB, por lo que el compresor (y eventualmente el descompresor) tiene un registro de cuáles fueron los últimos 32768 caracteres que fueron analizados. Asimismo, la longitud máxima considerada es de 258 bytes.

Durante el proceso de compactación, cuando la próxima secuencia de caracteres a compactar es idéntica a una que puede ser encontrada en la ventana deslizante, la secuencia de caracteres es reemplazada por dicho par de números: la distancia, que indica cuánto hay que retroceder en la ventana para encontrar el inicio de la secuencia, y la longitud, que determina la cantidad de caracteres que ambas secuencias comparten. En el caso en que la cadena no figure en la ventana deslizante, la cadena se ingresa directamente, es decir, no se produce compactación alguna.

Los valores de las distancias y las longitudes de cada par son almacenados en dos árboles de Huffman al comienzo de cada bloque del archivo de salida. Los bloques pueden ser de tamaño variable y se comienza uno nuevo cuando el tamaño de dichos árboles es excesivo.

BZIP2

Bzip2 es un compresor de archivos que utiliza básicamente dos algoritmos conocidos:

- El algoritmo de compresión de datos mediante el ordenamiento de bloques, sin pérdida de información, de Burrows-Wheeler.

- Codificación de Huffman.

La compresión lograda suele ser superior a la alcanzada por compresores más convencionales, basados en LZ77 o LZ78 y se acerca a la performance de los compresores estadísticos.

El mecanismo de compresión posee un overhead constante de 50 bytes y como mecanismo de seguridad para asegurar la fidelidad de los datos, un CRC de 32 bits (que asegura que la probabilidad de que no se detecte una corrupción de datos sea menor a 1 sobre 4 mil millones).

Bzip2 comprime archivos de tamaño considerable, mediante bloques. El tamaño del bloque afecta no sólo la proporción de compresión sino que también la cantidad de memoria necesaria para el proceso de compresión y descompresión. Mediante un flag del programa puede configurarse el tamaño de bloque entre 100.000 bytes y 900.000 bytes, alcanzándose una compresión superior eligiendo la última opción (en detrimento de una mayor utilización de memoria). Durante la compresión y descompresión, cada bloque se maneja de forma independiente; la representación comprimida de cada bloque está delimitada por un patrón de 48 bits, que permite identificar los límites de cada uno. Asimismo, cada bloque contiene como checksum un CRC de 32 bits.

LRZIP

Lrzip (Long Range Zip) es un compresor diseñado para comprimir archivos de gran tamaño (a partir de los 100MB). Este compresor está diseñado para obtener una mejor compresión de archivos grandes si cuenta con una gran cantidad de memoria RAM disponible.

El algoritmo del Lrzip se divide en dos etapas. En la primera etapa, busca y codifica fragmentos de texto duplicado sobre distancias muy largas en el archivo de entrada. Estas distancias para la búsqueda de bloques repetidos están limitadas por la cantidad de RAM disponible para el algoritmo. Es decir que cuando mayor es la memoria RAM disponible, este algoritmo será capaz de buscar redundancia dentro de ventanas más grandes. Es por este motivo que este compresor obtiene una mejor compresión a medida que se aumenta la memoria RAM disponible.

En la segunda etapa, utiliza otro compresor para comprimir la salida de la primera etapa. La implementación actual del Lrzip permite elegir entre el Lzma [15] o el Lzo [16] o el Bzip2 para comprimir la salida de la primera etapa. Lzma logra un

mayor radio de compresión que el Lzo y Bzip2 pero es más lento. En cambio el Lzo es el más rápido de los tres pero consigue compresiones más pobres.

La diferencia clave entre el Lrzip y los otros algoritmos de compresión mencionados aquí es su habilidad de poder encontrar redundancia de datos dentro de una ventana de gran tamaño. Por ejemplo, como ya hemos mencionado, el Bzip2 cuenta con una ventana para la búsqueda de repeticiones de 900kb mientras que el Gzip utiliza 32kb. En cambio, la ventana utilizada por el Lrzip está principalmente limitada por la cantidad de memoria RAM disponible.

GenCompress

GenCompress es un compresor [12] diseñado para comprimir secuencias de ADN. Es un algoritmo secuencial que utiliza un esquema similar al LZ77 [8]. La diferencia principal entre ambos es que LZ77 busca repeticiones exactas dentro de la "ventana deslizante", mientras que el GenCompress busca repeticiones aproximadas.

El algoritmo procede de la siguiente manera: Para una cadena de entrada w, asumimos que existen v y u subcadenas de w tal que $w=vu$, donde la subcadena v ya fue codificada y la subcadena u aún no lo fue. GenCompress busca un "prefijo óptimo" de u que se empareja de forma aproximada con alguna subcadena de v siempre que este prefijo pueda ser codificado de manera económica. Luego de escribir en la cadena de salida la codificación de este prefijo, lo remueve de u, y lo concatena como sufijo de v. Estos pasos se repiten hasta consumir toda la cadena u.

Debido a que el costo computacional de la búsqueda en cada paso del prefijo es óptimo, GenCompress utiliza un criterio de corte que limita la cantidad de operaciones de edición que puede tener cualquier subsecuencia dentro del prefijo. Por lo tanto la búsqueda del prefijo finaliza cuando la cantidad de operaciones es superada.

Como veremos más adelante, si bien GenCompress muestra muy buenos resultados en cuanto a la tasa de compresión, su performance no es buena.

Otros Compresores

En la actualidad existen otros compresores específicos para ADN. Entre ellos se destaca el GenML ya que presenta los mejores resultados para compresión de secuencias de ADN. Pero sus autores, hasta el momento, solo dieron a conocer los resultados del compresor para los dos datasets que suelen utilizarse como

benchmark para compresores de ADN, los archivos comprimidos y un descompresor para que la comunidad científica pueda verificar la existencia del mismo. Por este motivo este compresor no formó parte de las pruebas realizadas en este trabajo.

Resultados experimentales

Con el objetivo de corroborar si los compresores utilizados en este trabajo cumplen con las propiedades de compresores normales realizamos una serie de pruebas con 6 datasets de ADN. Estas pruebas consistieron en verificar qué compresores cumplen con las propiedades de idempotencia a) y de simetría c). Además también sirvieron para demostrar de manera empírica si estas propiedades se mantienen inmutables con respecto al tamaño de los archivos.

Los compresores utilizados en estas pruebas son:

- Bzip2
- GenCompress
- Lrzip
- Gzip

Cada dataset utilizado está formado por un conjunto de archivos, donde cada archivo contiene una secuencia de ADN.

El dataset 1 y el dataset 2 son los datasets estándares [12] utilizados para comparar a los compresores de ADN. En particular el dataset 1 incluye el genoma completo de dos mitocondrias: MPOMTCG, PANMTPACGA; dos cloroplastos: CHNTXX y CHMPXX; cinco secuencias humanas: HUMGHCSA, HUMHBB, HUMHDABCD, HUMDYSTROP, HUMHPRTB; y finalmente dos genomas completos de dos virus: VACCG y HRHCMVVCG. Mientras que el dataset 2 incluye las secuencias de ADN: CELK07E12, ATEF1A23, HSG6PDGEN, ATRDNAI, ATATSGS, XLXFG512, ATRDNAF y MMZP3G.

	Dataset 1	***Dataset 2***	***Dataset 3***	***Dataset 4***	***Dataset 5***	***Dataset 6***
Nro. de muestras	10	8	5	6	5	5
Longitud Promedio	118 kb	21 kb	22 Mb	53 Mb	84 Mb	158 Mb
Desviación est.	65 kb	21 kb	13 Mb	2 Mb	294 kb	26 Mb

Tabla 4 – Propiedades de los datasets

Los datasets 2, 3, 4, 5 y 6 están compuestos por cromosomas de 5 especies donde cada uno de ellos contiene muestras de tamaño similar (Tabla 4). Las secuencias de estos dataset fueron obtenidas del proyecto Ensembl [17]. En la

siguiente tabla se muestra cuales son los cromosomas que componen cada dataset (Tabla 5).

	Dataset 3	*Dataset 4*	*Dataset 5*	*Dataset 6*
Pan troglodytes (Chimpancé)	Y	20	13	3
Homo Sapiens (Humano)	Y	19	14	5
Canis familiaris (Perro)	35	19	4	1
Rattus norvegicus (Rata)	12	19	17	3
Mus musculus (Ratón)	Y	19	18	2

Tabla 5 –Cromosoma de cada especie por dataset

Análisis de la capacidad de compresión

Para comparar la capacidad de compresión de los compresores utilizaremos el concepto de factor de compresión:

$$Factor\ De\ Compresión = \frac{Tamaño\ Cadena\ Entrada}{Tamaño\ Cadena\ Salida}$$

En este caso, los valores mayores a 1 indican que se comprimió la cadena de entrada, mientras que valores menores a 1 indican que el compresor expandió la cadena de entrada. Esta medida es intuitiva ya que mayor es el factor, mayor es la compresión obtenida, por ende, mejor el compresor.

En la Tabla 6 podemos observar el factor de compresión promedio obtenidos por los compresores para cada dataset. El GenCompress es el compresor que mejor resultados consiguió para los datasets 1 y 2, pero no fue posible comprimir los archivos de los datasets 3, 4, 5 y 6 porque no terminaba de procesarlos. El Gzip obtuvo el peor factor de compresión en todos los datasets. Esto era esperable porque es el compresor que tiene la ventana deslizante más chica. Por este mismo motivo el Lrzip es el que mejor factor de compresión logró para los datasets de mayor tamaño promedio (3, 4, 5 y 6), ya que utiliza toda la memoria disponible. También se puede corroborar que el GenCompress comprime mejor que el Lrzip ya que busca subsecuencias aproximadas y reversos complementarios.

	Gzip	*Bzip2*	*GenCompress*	*Lrzip*
Dataset 1	3.71	3.84	**4.74**	4.11
Dataset 2	3.88	3.94	**4.96**	4.06
Dataset 3	3.80	3.99	-	**5.38**
Dataset 4	3.80	3.92	-	**4.40**
Dataset 5	3.71	3.86	-	**4.38**
Dataset 6	3.70	3.86	-	**4.39**

Tabla 6 – Comparación de factor de compresión promedio

En este experimento tratamos de buscar cuál es el límite del GenCompress en relación al tamaño del archivo de entrada que puede comprimir finalizando exitosamente en un tiempo razonable.

Análisis de tiempo de ejecución

La Figura 3 muestra el tiempo de ejecución promedio que necesitó cada compresor para comprimir los archivos de los datasets 1 y 2. Como podemos observar el GenCompress es el compresor de peor rendimiento. En particular, para el dataset 2 tardó, en promedio, 10 veces más en procesar cada archivo que el resto de los compresores.

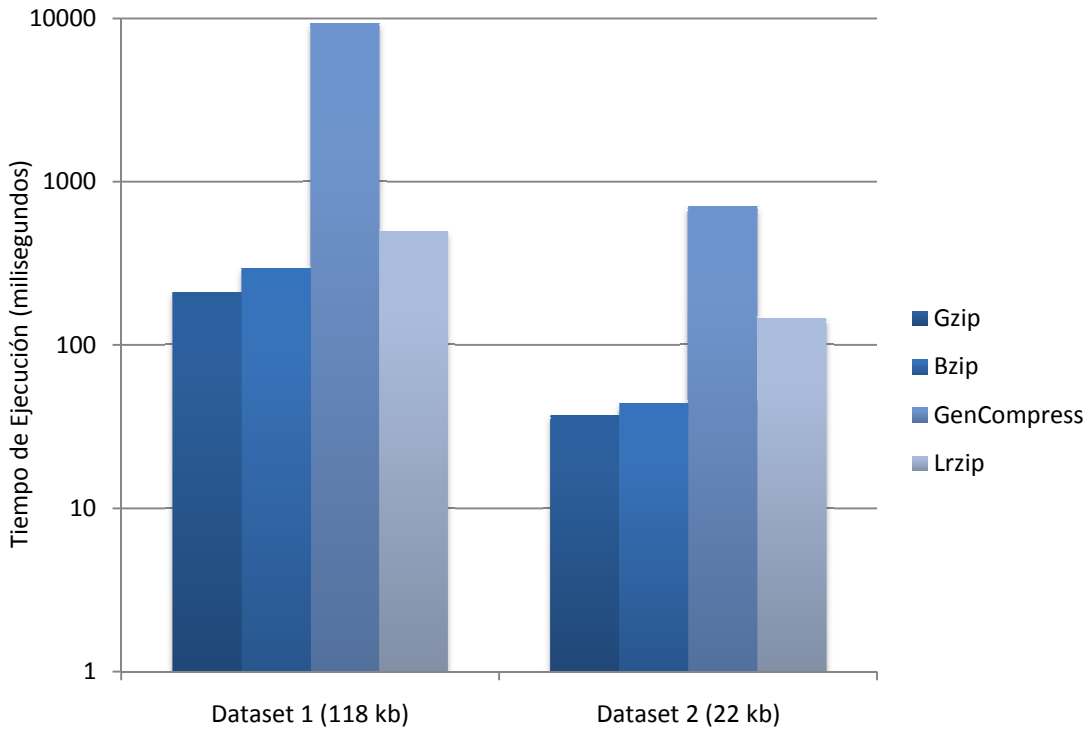


Figura 3 - Tiempo de ejecución promedio de los datasets 1 y 2

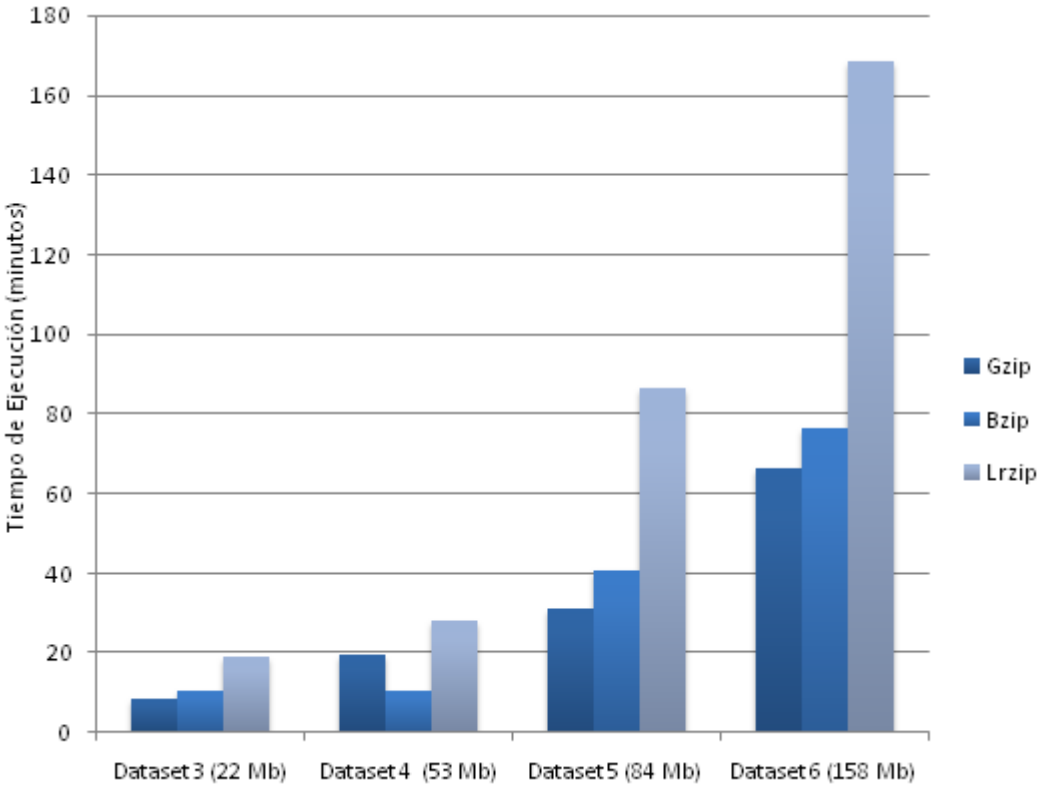


Figura 4 - Tiempo de ejecución promedio Dataset 3, 4, 5 y 6

La Figura 4 muestra el tiempo de ejecución promedio que necesitó cada compresor para comprimir los archivos de los datasets 3, 4, 5 y 6. Teniendo en cuenta lo mencionado en el "Análisis de la capacidad de compresión", podemos sostener que los compresores que mejor comprimen son los que tardan más tiempo en ejecutar.

Análisis de la propiedad de idempotencia

Para analizar si los compresores utilizados en este trabajo cumplen con la propiedad de idempotencia a), definimos el Coeficiente de Idempotencia C_I de la siguiente manera:

$$C_I(x) = 1 - \frac{C(xx) - C(x)}{C(x)}$$

$C(x)$ especifica el tamaño resultante de comprimir el archivo x. Por lo tanto valores cercanos a 1 en este coeficiente implican que $C(x) \approx C(xx)$, mientras que

valores cercanos a 0 implican que $C(xx) \approx 2\,C(x)$. Por lo tanto los valores cercanos a 1 implican que el compresor cumple con la propiedad de idempotencia, mientras que valores cercanos a 0 implican que no la cumple.

	Gzip	***Bzip2***	***GenCompress***	***Lrzip***
Dataset 1	0.56	0.05	**1.00**	**1.00**
Dataset 2	0.63	0.73	**1.00**	**0.98**
Dataset 3	0.00	0.00	-	**0.80**
Dataset 4	0.00	0.00	-	**0.02**
Dataset 5	0.00	0.00	-	**0.01**
Dataset 6	0.00	0.00	-	0.00

Tabla 7 - Coeficiente de idempotencia promedio

La Tabla 7 muestra el coeficiente de idempotencia promedio obtenido por cada compresor en cada dataset. El GenCompress cumple con la propiedad de idempotencia para los dos datasets en donde se lo probó, por lo tanto podemos sostener que es capaz de encontrar repeticiones dentro de una ventana de 236 Kb (2 x 118 Kb). El Lrzip cumple con la propiedad de idempotencia en los datasets 1, 2 y 3, por lo tanto podemos sostener que es capaz de encontrar repeticiones dentro de una ventada de 44 Mb (2 x 22 Mb).

En cuanto a los compresores Gzip y Bzip2 podemos decir que no cumplen la propiedad de idempotencia en ningún dataset. Lo cual era esperable, debido a sus actuales implementaciones y al tamaño de los archivos de los datasets.

Análisis de la propiedad de simetría

Para analizar la propiedad de simetría c), definimos el Coeficiente de Simetría C_S entre dos secuencias de la siguiente manera:

$$C_S(x, y) = |\, C(xy) - \, C(yx) \,|$$

Este coeficiente se calcula como el valor absoluto de la diferencia entre la longitud de la compresión de la concatenación xy y la longitud de la compresión de la concatenación yx. En este caso decimos no normalizar este coeficiente ya prácticamente la diferencia entre $C(xy)$ y $C(yx)$ era muy pequeña en relación a la longitud de la concatenación de las cadenas x e y.

	Gzip	***Bzip2***	***GenCompress***	***Lrzip***
Dataset 1	77	4	52	187
Dataset 2	31	0	10	43
Dataset 3	659	7832	-	9583
Dataset 4	363	7302	-	12710
Dataset 5	371	7874	-	6501
Dataset 6	484	**17527**	-	7665

Tabla 8 - Coeficiente de simetría promedio en bytes

La Tabla 8 muestra el coeficiente de simetría promedio obtenido por cada compresor en cada dataset. Se puede observar que todos los compresores cumplen con la propiedad de simetría c) dado los valores obtenidos de C_S son despreciables en relación de la longitud de los archivos de cada datasets. Por ejemplo, el peor caso lo presenta el Bzip2 con el Dataset 6, donde en promedio el $C_S(x, y)$ es de 17527 bytes. Pero si tenemos en cuenta que la longitud promedio de los archivos del Dataset 6 es de 158 Mb, y por ende la longitud promedio de la concatenación de los archivos asciende a 316 Mb (2 x 158 Mb) aproximadamente, podemos sostener que 17522 bytes de diferencia entre $C(xy)$ y $C(yx)$ no es un valor significativo.

NCD aplicada a la filogenia

El principal objetivo de este trabajo consiste en estudiar el uso de la *distancia de compresión normalizada* (NCD) propuesta por Vitányi cómo métrica para comparar secuencias biológicas.

En biología existen varios problemas que se resuelven computacionalmente comparando secuencias biológicas. Uno de estos problemas es el de inferir árboles filogenéticos. Como vimos en la sección anterior existen varios enfoques para inferir este tipo de árboles. Uno de ellos se basa en matrices de distancia. Como explicamos anteriormente, para generar estas matrices es necesario definir una métrica que dadas dos secuencias establezca una noción de similitud entre ellas.

Nuestro trabajo consistió en utilizar la NCD para calcular matrices de distancias que junto a un método de clustering generarán árboles filogenéticos. Para cotejar la utilidad de este método, comparamos los árboles filogenéticos obtenidos con nuestro método con los árboles filogenéticos generados por los métodos de distancia ya implementados en PAUP. Además de comparar nuestros resultados con los obtenidos métodos de distancia, también realizamos comparaciones contra los métodos de máxima parsimonia y máxima verosimilitud. Para realizar estas comparaciones se utilizó la métrica de comparación de árboles binarios llamada *agreement*.

A continuación se describirá la métrica agreement. Luego se brindará una breve introducción al PAUP y finalmente se explicará de manera detallada la aplicación que desarrollamos para la inferencia de árboles filogenéticos basados en NCD y su posterior comparación con los árboles generados en PAUP.

Introducción a la métrica Agreement

Para poder comparar los árboles filogenéticos generados con nuestro método con los árboles filogenéticos obtenidos con los métodos de distancia implementados en PAUP necesitamos buscar un método de comparación de árboles binarios.

En [6], Goddard desarrolló un algoritmo de orden cuadrático de comparación de árboles binarios llamado *Agreement*. Como los árboles filogenéticos son binarios y este algoritmo ya fue utilizado para este propósito [19], decidimos utilizarlo para comparar nuestros resultados con los obtenidos en PAUP.

El algoritmo de Agreement define la operación de *poda*. Esta operación consiste en remover una o más hojas del árbol y eliminar todos los nodos internos de grado dos que fueron generados a partir de la eliminación de las hojas. Por ejemplo, la Figura 5 muestra el árbol T_2 resultante de aplicar la operación de *poda* a la hoja 2 del árbol T_1.

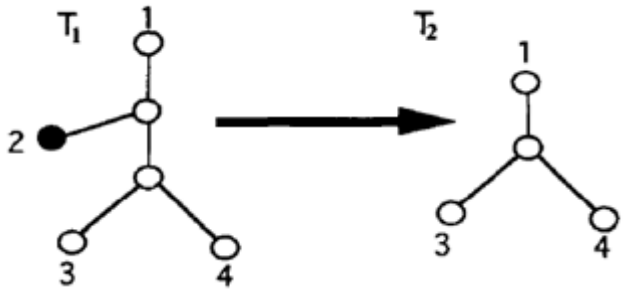


Figura 5 – Resultado de la operación de poda del nodo 2

Dados dos árboles binarios *T* y *U*, se define al *subárbol agreement* como el árbol que puede ser obtenido desde *T* o U realizando la menor cantidad de operaciones de *poda*.

El concepto de *subárbol agreement* fue motivado por el objetivo de formalizar una noción de similitud entre dos árboles binarios. En base a este concepto se define la distancia agreement d_A entre los árboles binarios T y U de n hojas de la siguiente manera:

$$d_A: Arbol\ Binario\ \times Arbol\ Binario\ \rightarrow R$$
$$d_A(T,U) = n - \#A(T,U)$$

Donde $A(T,U)$ es el conjunto de todos los subárboles agreement existentes entre T y U, mientras que $\#A(T,U)$ denota su tamaño. Esta métrica determina la menor cantidad de hojas que deben ser podadas de los árboles T y U para obtener un subárbol en común.

En la Figura 6 se muestra un ejemplo de esta métrica aplicada a los árboles T_1 y T_2. En este caso, la distancia agreement entre ambos es 1 ya que basta con aplicar una operación de *poda* sobre el nodo x para obtener el subárbol agreement.

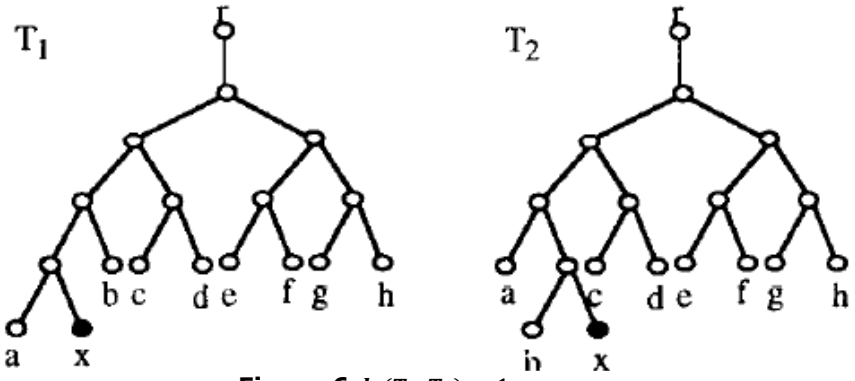


Figura 6 $d_A(T_1, T_2) = 1$

PAUP

PAUP [28] es un software que contiene diversos métodos para la inferencia de árboles filogenéticos. En particular, cuenta con un conjunto de algoritmos para la generación de árboles filogenéticos basado en distancias. En total PAUP puede utilizar 17 métricas para calcular las matrices de distancias, dentro de las cuales se destacan las siguientes: JC[34], F81[35], TAJNEI[36], K2P[37], F84[38], HKY85[39], K3P[40], TamNei[41], K2P[42], GTR[43][44] y LogDet[44][45]. En general estas distancias determinan la similitud entre dos secuencias de ADN aplicando diferentes modelos estadísticos (tasas de sustituciones, frecuencias) sobre los nucleótidos de las secuencias a comparar.

PAUP provee dos tipos de interfaces: una gráfica y una de líneas de comandos. Ambas interfaces exponen las mismas funcionalidades, aceptan los mismos tipos de datos de entrada y ejecutan las mismas operaciones.

La Figura 7 –muestra la interfaz gráfica de PAUP 4.0b10. La parte inferior de la pantalla contiene la una línea de comandos, mientras que la parte superior muestra los resultados de las operaciones ejecutadas en dicha línea.

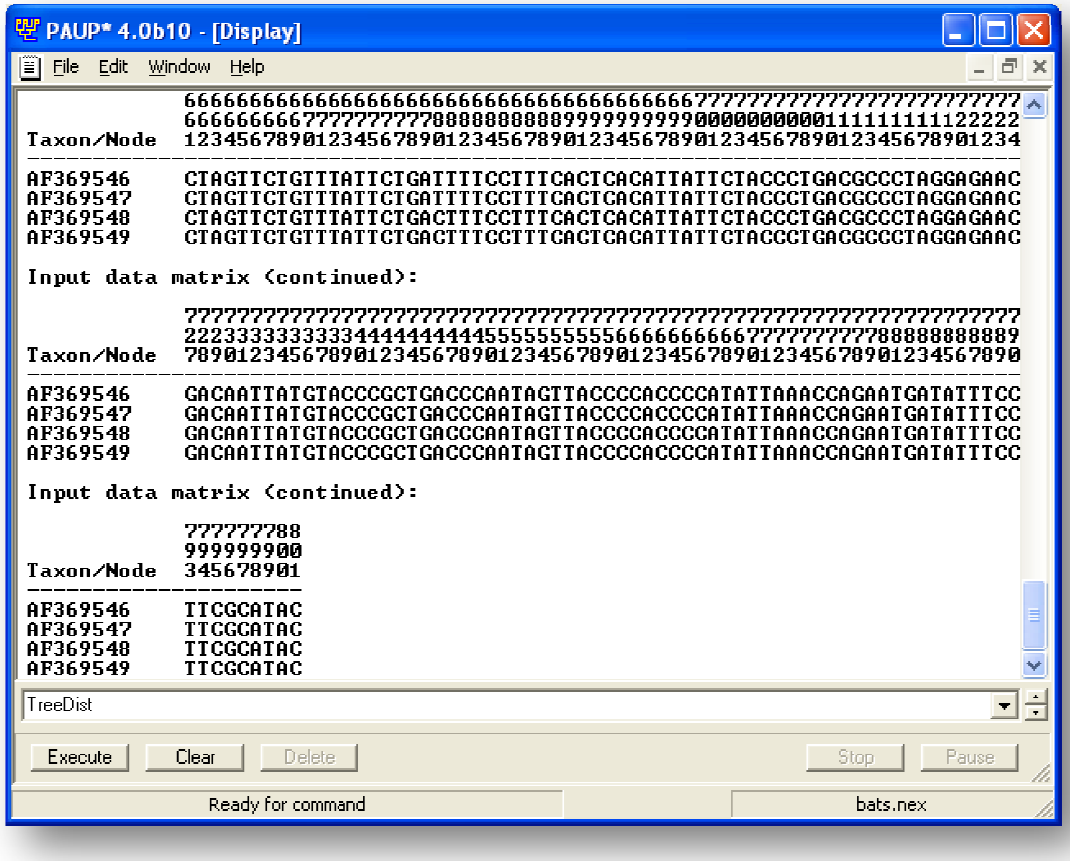


Figura 7 – Interfaz gráfica del PAUP

PAUP cuenta con un lenguaje de scripting llamado NEXUS. Este lenguaje permite especificar comandos para la importación de datasets, la ejecución de operaciones y la visualización o escritura de los resultados. El usuario puede decidir entre ejecutar operación por operación utilizando la interfaz gráfica o bien escribir un script en formato NEX y luego ejecutarlo con el motor de PAUP. En nuestro caso, la interacción con PAUP se realiza a través de scripts NEXUS.

Inferencia de árboles filogenéticos

En esta sección se describirá de manera detallada cada uno de los pasos realizados por la aplicación que fue desarrollada en el contexto de este trabajo, con el objetivo de poder generar árboles filogenéticos utilizando la métrica NCD para luego compararlos con los árboles generados utilizando los distintos métodos de PAUP. En la Figura 8 se muestra, de manera esquemáticamente, los diferentes pasos ejecutados por nuestra aplicación para cumplir con el objetivo propuesto.

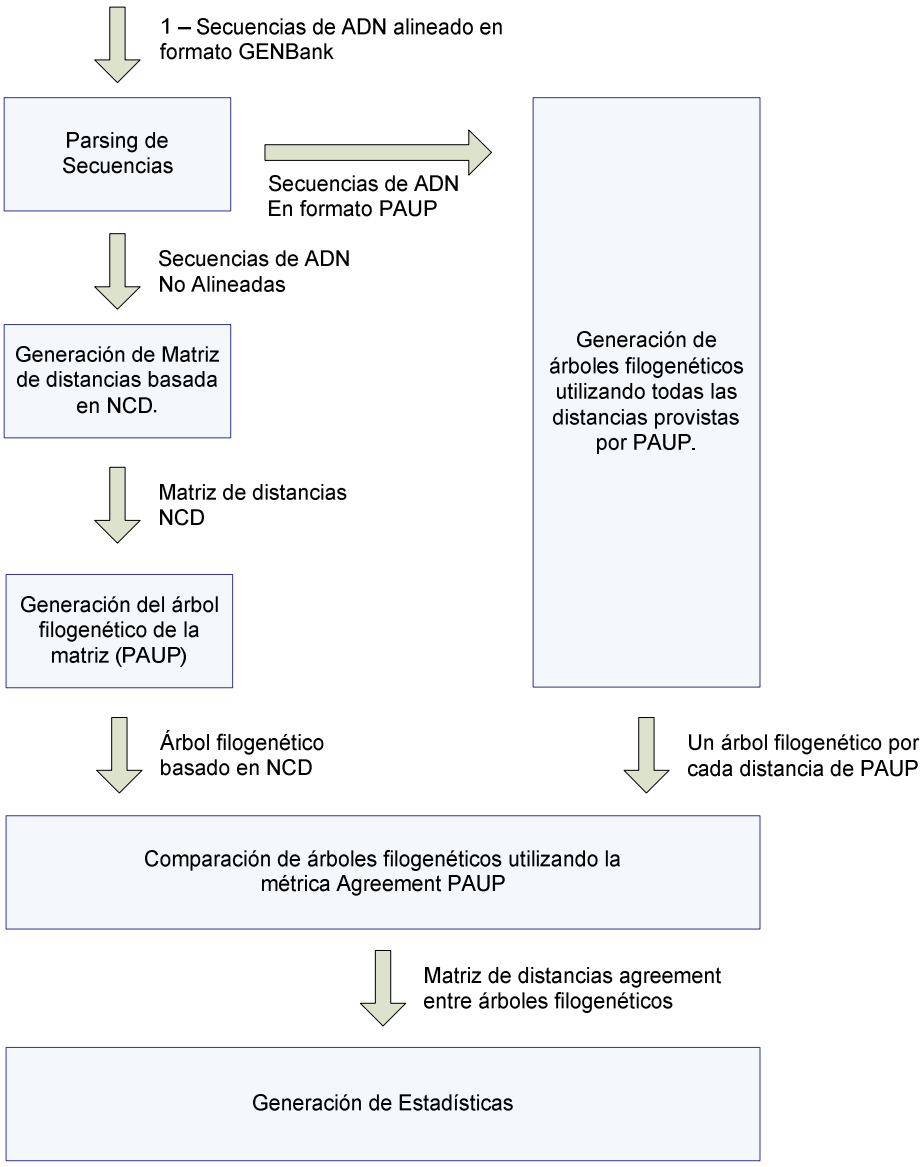


Figura 8 - Etapas de la aplicación

Datasets

La entrada de nuestra aplicación consiste en un conjunto de datasets con secuencias de ADN (no necesariamente alineadas). Cada uno de estos datasets está constituido por un archivo que contienen entre quince y cuarenta secuencias de ADN mitocondrial. Al final del proceso, para cada uno de estos datasets se construirá un árbol filogenético utilizando la métrica NCD.

En la Figura 9 se muestra la parte inicial del archivo de secuencias del dataset *AfricanHominoids* en el formato de GenBank [22]. La parte superior contiene una descripción de las secuencias de ADN y el trabajo donde se utilizaron dichas secuencias. Dicha descripción es seguida con el contenido de doce secuencias de ADN alineadas[2].

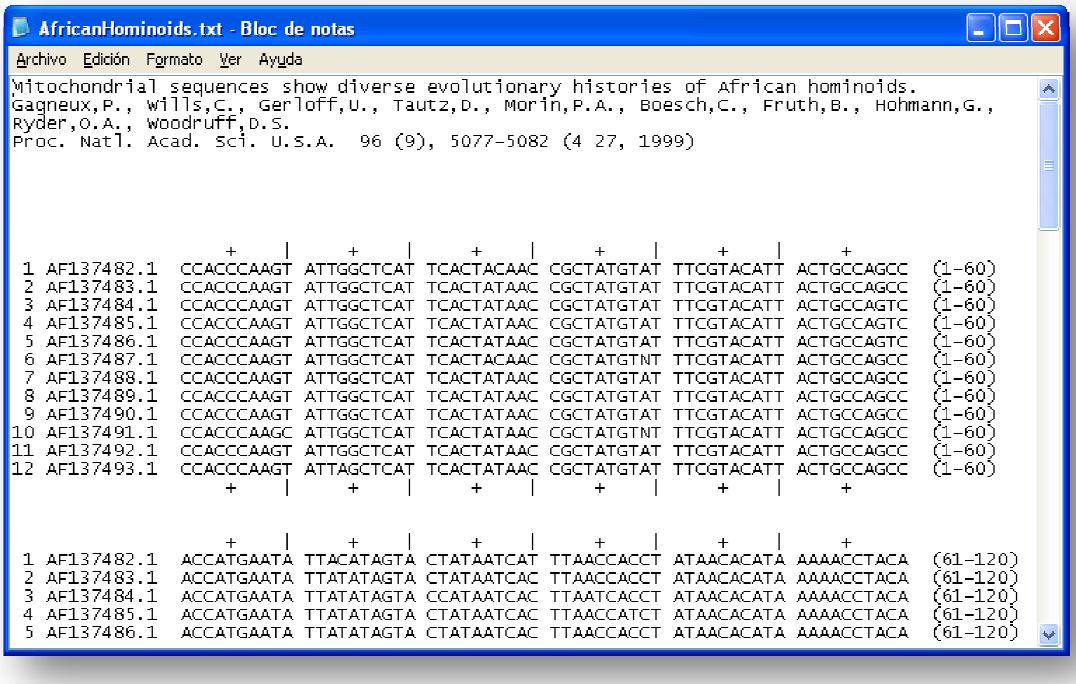


Figura 9 – Archivo con secuencias en formato GenBank

Preparación de los datos

En el segundo paso se realiza un pre-procesamiento de las secuencias de ADN. Como describimos anteriormente, una característica fundamental de nuestro método que es no requiere que las secuencias de ADN se encuentren alineadas, a

[2] Algunos Datasets están constituidos por secuencias previamente alineadas. Cómo se verá a continuación, en esos casos, procederemos a desalinear, a fin de poder verificar el método propuesto en este trabajo.

diferencia de la mayoría de los métodos de generación de árboles filogenéticos. La mayoría de los datasets utilizados en este trabajo, proceden del GenBank, y contienen secuencias alineadas. Dado que nuestro objetivo es usar secuencias no alineadas, en esta etapa se procede a desalinear cada conjunto de secuencias almacenando cada secuencia sin alinear en un archivo diferente con el nombre de la secuencia. Paralelamente, las secuencias de ADN alineadas son transformadas al formato requerido por PAUP y almacenadas en un archivo con extensión NEX. Las secuencias no alineadas serán utilizadas por nuestro método, las secuencias alineadas en formato NEX serán directamente utilizadas con PAUP (PAUP requiere que las secuencias se encuentren alineadas). Finalmente, compararemos los resultados.

Siguiendo con el ejemplo del dataset *AfricanHominoids*, luego del pre-procesamiento inicial obtendremos, por un lado, las doce secuencias de ADN no alineadas almacenadas en archivos separados (Figura 10), y por otro lado, guardaremos las doce secuencias de ADN alineadas en un único archivo con formato .NEX (Figura 11).

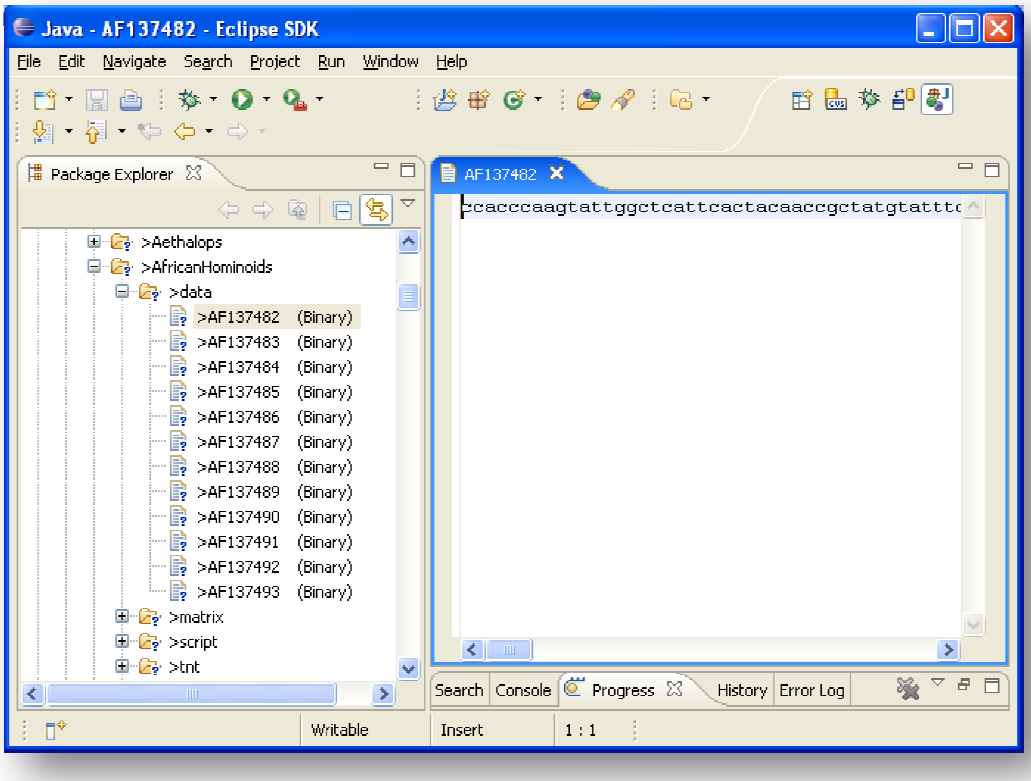


Figura 10 - Secuencias de ADN no alineadas

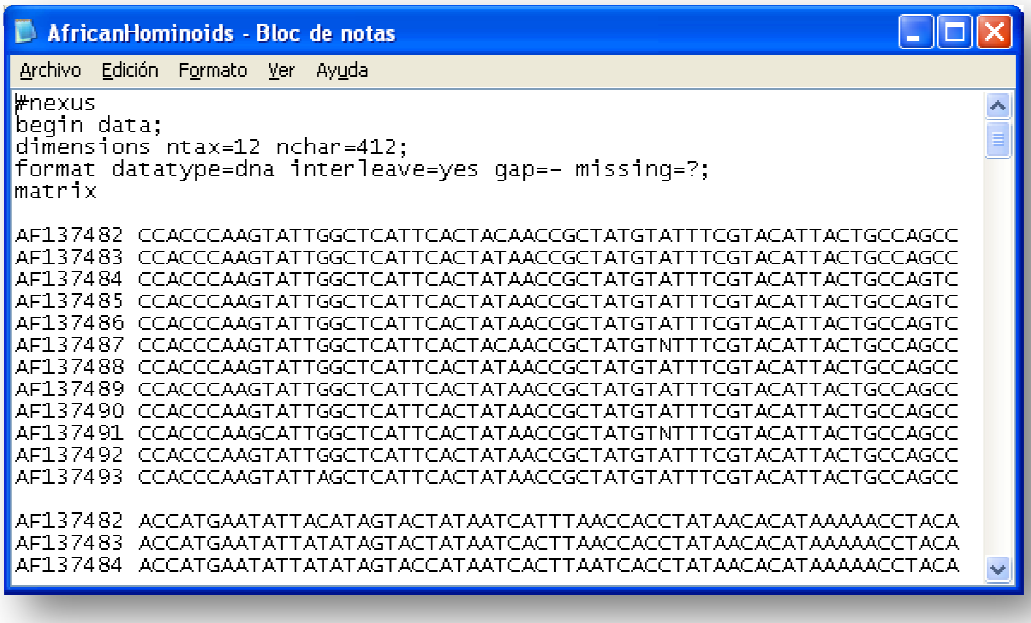


Figura 11 - Dataset en formato NEX utilizado por PAUP

Procesamiento en PAUP

En este paso, la aplicación construirá los árboles filogenéticos asociados a las secuencias de ADN, utilizando los métodos de distancia provistos por PAUP. Mediante la utilización de scripts, primero se importarán las secuencias de ADN, luego se ejecutarán los comandos necesarios para poder generar los árboles filogenéticos utilizando los distintos métodos de distancia de PAUP y finalmente se almacenarán los árboles generados en un archivo de salida. Este archivo de salida está escrito en formato .NEX y será utilizado más adelante en el proceso de validación de la métrica NCD.

En la Figura 12 se muestra un árbol filogenético del dataset *AfricanHominoids* creado por PAUP utilizando como matriz de distancias la métrica F84 [38].

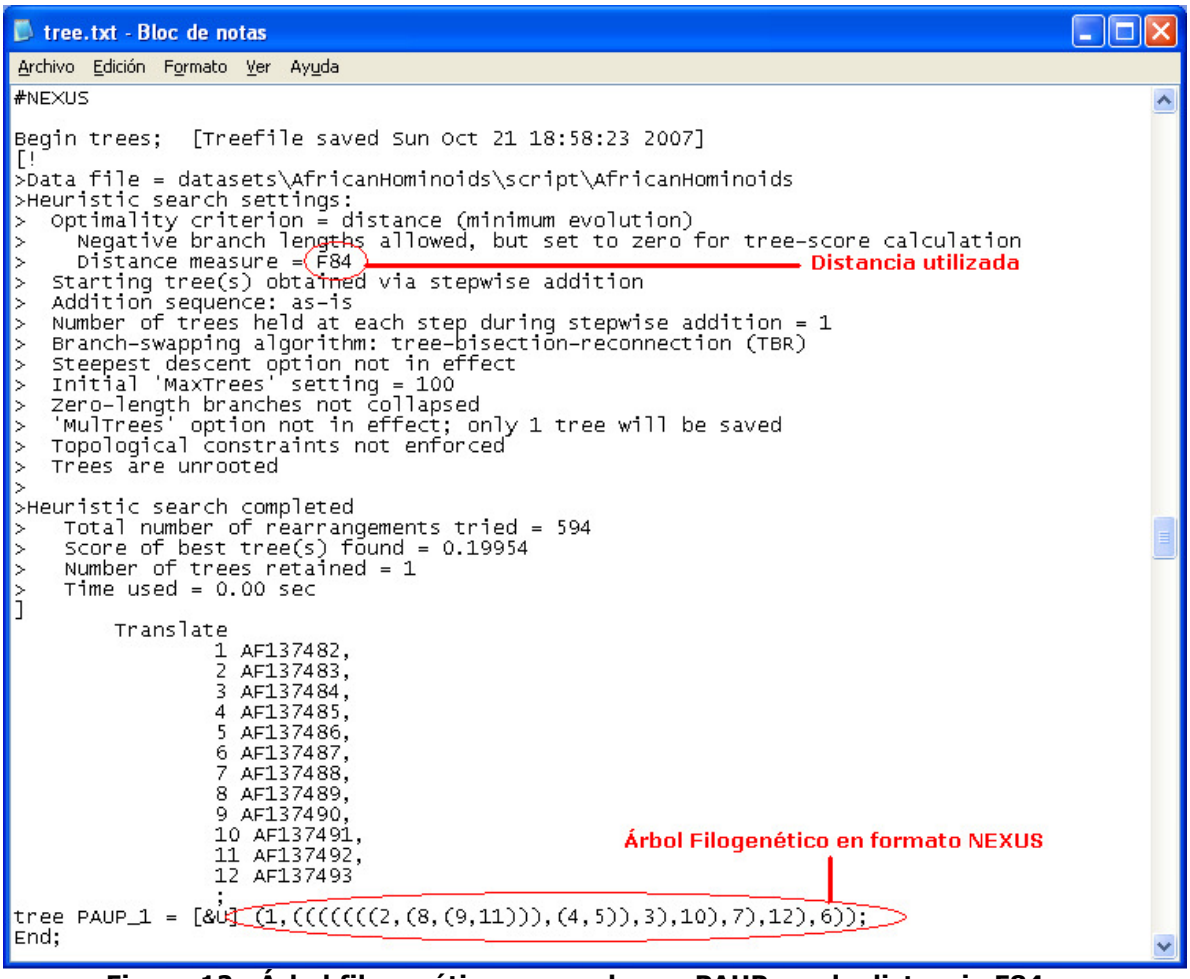


Figura 12 –Árbol filogenético generado por PAUP con la distancia F84

Matriz de distancia NCD

En esta etapa se ejecuta nuestro algoritmo de generación de árboles filogenéticos. Como primer paso se construye la matriz de distancias M utilizando la métrica NCD. En esta matriz se almacenarán las distancias calculadas entre todas las secuencias de ADN.

Cada posición de la matriz se calcula de la siguiente manera:

$$M[i,j] = \frac{C(S_i + S_j) - \min(C(S_i) + C(S_j))}{\max(C(S_i) + C(S_j))}$$

Donde:

- S_k representa a la secuencia de ADN k.
- $S_i + S_j$ representa la concatenación de las secuencias de ADN S_i y S_j.
- $C(S_k)$ representa a la compresión de la secuencia S_k

• A continuación se muestra un pseudo-código de la generación de la matriz de distancias M que será útil para calcular el orden de ejecución del algoritmo.

```
Sea S : Array of Secuencias de ADN [0..N]
Sea G : Array of Double [0..N]

Function GenerarMatriz ()

  For i = 0 to N - 1

    For j = i + 1 to N - 1

      CompS_i = getTamañoCompresion(i)
      CompS_j = getTamañoCompresion(j)

      // Concatenación de las secuencias S(i) y S(j).
      // O(k+t), k = tamaño de S(i), t = tamaño de S(j)
      CompConjunta = Comprimir(S(i) + S(j))

      norm = min(CompS_i, CompS_j) / max (CompS_i, CompS_j)

      M[i, j] = CompConjunta - norm

    End For

  End For

  Return M;

End Function

Function getTamañoCompresion(int idSecuencia)

  IF (G(idSecuencia) != null) THEN

    // Compresion de la secuencia S(i).
    G(idSecuencia) = Comprimir(S(i))       // O(k), k = tamaño de S(i)

  END IF

  Return G(idSecuencia)

End Function
```

Como podemos observar el orden de complejidad algorítmica para el cálculo de la matriz M depende del orden del orden de complejidad del compresor elegido. Por lo tanto el orden de ejecución de este algoritmo es $O(n^2 \cdot f_c(k))$, donde n es la cantidad de secuencias de ADN, k es la cantidad de caracteres que contiene la secuencia de ADN más larga y f_c es la complejidad del compresor c.

Utilización de la NCD en PAUP

A partir de la matriz de distancia constituida en el punto anterior, el siguiente paso es construir el árbol filogenético correspondiente. Para ello utilizaremos la implementación del algoritmo de clustering Neighbor Joining (NJ) [18] que utiliza PUAP. Este algoritmo es el mismo algoritmo que usa PAUP para generar los árboles filogenéticos basados en matrices de distancia.

Para generar al árbol filogenético basado en la NCD, primero se debe importar en PAUP la matriz de distancia generada y luego, mediante un script, ejecutar el algoritmo de clustering NJ.

En el siguiente gráfico se muestra el script ejecutado en PAUP para construir el árbol filogenético del dataset *AfricanHominoids* en base a la matriz de distancia generada con la métrica NCD.

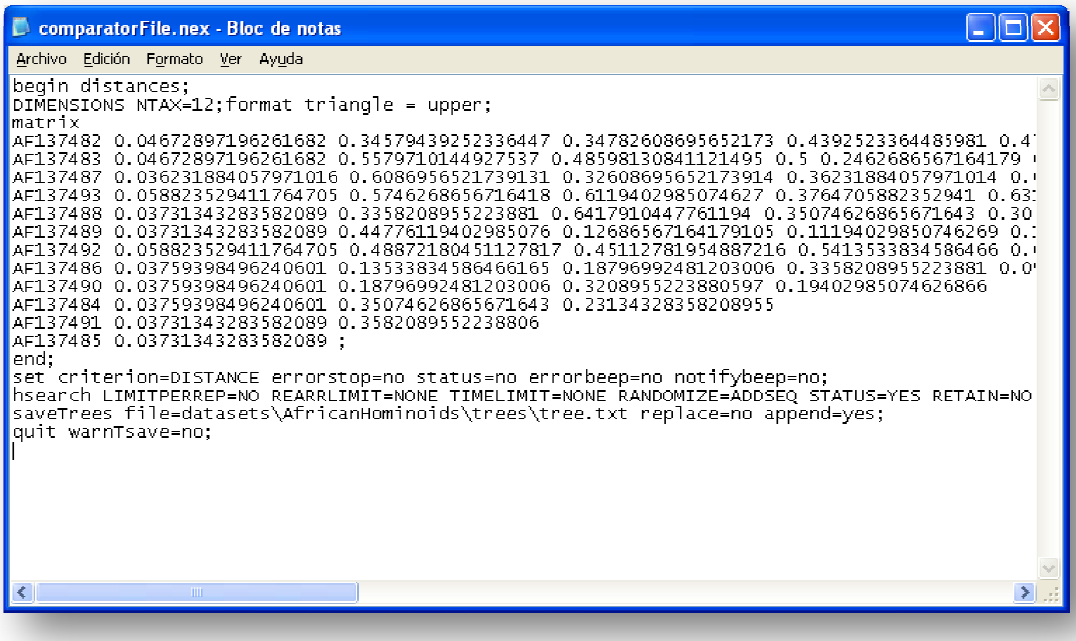


Figura 13 – Script para calcular el árbol filogenético en PAUP usando NJ en base a la matriz de distancia NCD

Validación de la métrica NCD

Por último debemos comparar el árbol filogenético basado en la métrica NCD con los demás árboles filogenéticos construidos utilizando las distancias de PAUP. Para ello, utilizaremos la métrica Agreement que ya hemos introducido anteriormente [6]. PAUP brinda la funcionalidad de comparación de árboles utilizando esta medida.

Mediante un script, se importan en PAUP todos los árboles generados en los pasos anteriores. Luego ejecuta el método de comparación Agreement. El resultado de esta comparación será una matriz que detallará la distancia existente entre todos los árboles generados, a partir de la cual calcularemos nuestros resultados finales.

En la Figura 14 se muestra el script utilizado para obtener las distancias de los árboles generados para el dataset *AfricanHominoids*. En la parte superior del mismo se muestra el último de los árboles filogenéticos generados. Luego se ejecuta el comando *treeDist* especificando la métrica Agreement.

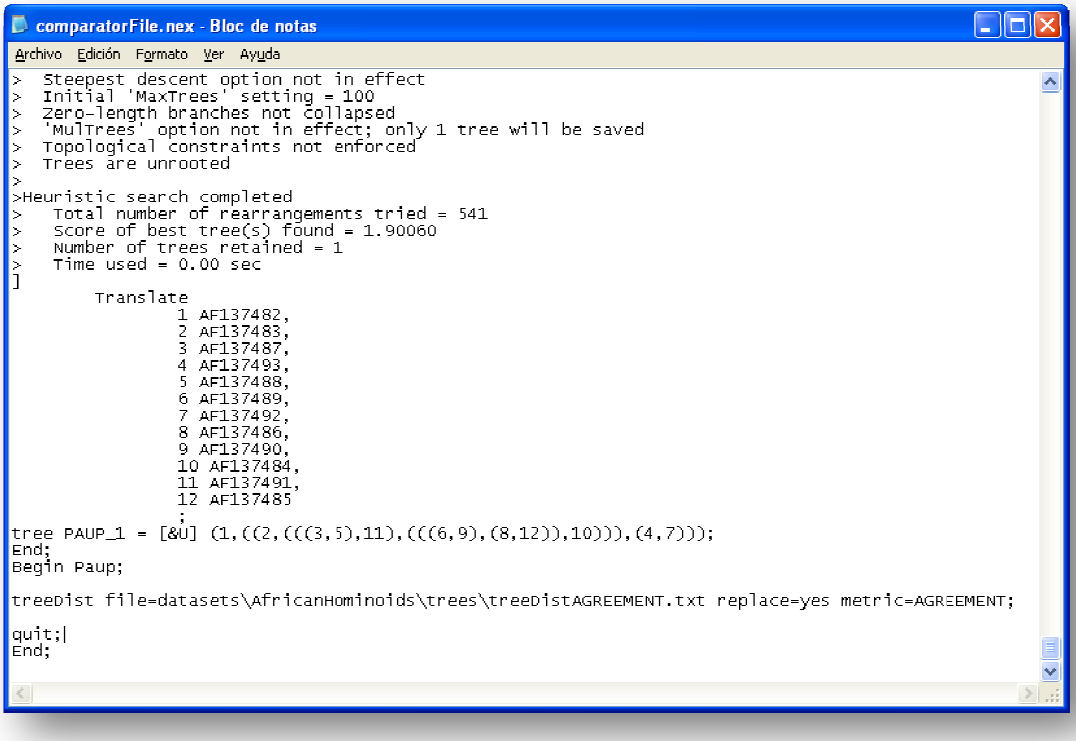
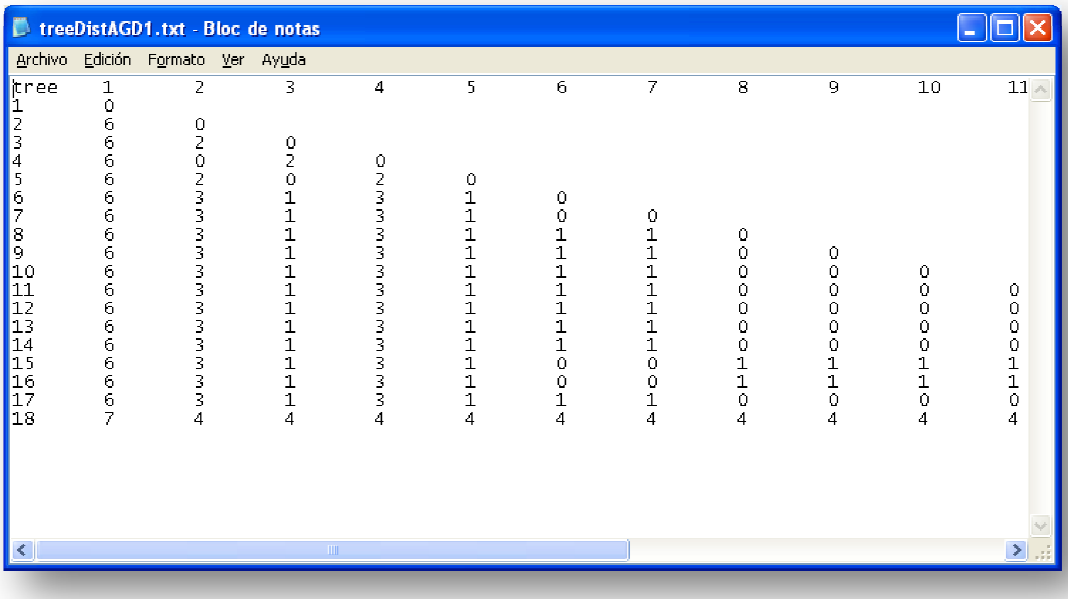


Figura 14 – Script para calcular la distancia Agreement entre todos los árboles generados

Figura 15 se muestra el archivo resultante de la operación treeDist. Este archivo contiene la matriz con las distancias Agreement calculada entre todos los árboles filogenéticos generados en los pasos anteriores.



tree	1	2	3	4	5	6	7	8	9	10	11
1	0										
2	6	0									
3	6	2	0								
4	6	0	2	0							
5	6	2	0	2	0						
6	6	3	1	3	1	0					
7	6	3	1	3	1	0	0				
8	6	3	1	3	1	1	1	0			
9	6	3	1	3	1	1	1	0	0		
10	6	3	1	3	1	1	1	0	0	0	
11	6	3	1	3	1	1	1	0	0	0	0
12	6	3	1	3	1	1	1	0	0	0	0
13	6	3	1	3	1	1	1	0	0	0	0
14	6	3	1	3	1	1	1	0	0	0	0
15	6	3	1	3	1	0	0	1	1	1	1
16	6	3	1	3	1	0	0	1	1	1	1
17	6	3	1	3	1	1	1	0	0	0	0
18	7	4	4	4	4	4	4	4	4	4	4

Figura 15 – Matriz de distancia entre los árboles filogenéticos generados por cada distancia

Comparación

Finalmente, tomando todas las matrices de distancia Agreement generadas con los distintos datasets, construimos una tabla comparativa con los resultados obtenidos. Cada columna de la tabla representa una distancia de PAUP y cada fila representa un dataset. Entonces una celda de la tabla contiene la distancia Agreement entre el árbol filogenético generado con la métrica NCD y el árbol filogenético de la distancia de PAUP especificado por la columna para el dataset especificado por la fila.

Siguiendo con nuestro ejemplo del dataset AfricanHominoids, en la Figura 16, podemos ver que la distancia entre el árbol filogenético calculado con la distancia ABS (PAUP) y la distancia NCD es 4. En la siguiente sección se explicará la interpretación de estos resultados.

Dataset	# Taxas	Best SymDiff(#)	Default			TOTAL			MEAN			ABS		
Aethalops	21	16(4)	34	12	12.273810	16	7	7.204082	16	7	7.204082	16	7	7.204082
AfricanHominoids	12	16(16)	18	7	7.404762	16	4	4.416667	16	4	4.458333	16	4	4.416667
AfricanTilapiineFishes	12	2(2)	18	6	6.486111	4	2	2.333333	4	2	2.333333	4	2	2.333333
AmericanCollaredLemmings	17	4(1)	28	10	10.347059	6	2	2.264706	8	3	3.254902	6	2	2.264706
AncientCetaceanlineages	9	6(2)	12	4	4.472222	8	3	3.444444	8	3	3.444444	8	3	3.444444
AncientCetaceanlineages2	6	2(16)	4	1	1.833333	2	1	1.666667	2	1	1.666667	2	1	1.666667
AncientCetaceanlineages3	6	0(16)	6	2	2.666667	**0**	**0**	**0.000000**	**0**	**0**	**0.000000**	**0**	**0**	**0.000000**
angiosperms	13	16(16)	20	6	6.410256	16	6	6.371795	16	6	6.371795	16	6	6.371795
angiosperms2	17	0(16)	28	8	8.448529	**0**	**0**	**0.000000**	**0**	**0**	**0.000000**	**0**	**0**	**0.000000**
anopheles-arabiensis	18	4(15)	30	11	11.292929	4	3	3.222222	4	3	3.222222	4	3	3.222222
AnophelesOswaldoi	25	6(1)	44	18	18.204444	8	4	4.160000	8	4	4.160000	8	4	4.160000
AnophelesPunctulatus	11	4(15)	16	6	6.409091	4	2	2.363636	4	2	2.363636	4	2	2.363636
anophelinae2	19	16(4)	32	10	10.363158	16	6	6.254386	16	6	6.254386	16	6	6.254386
AntelopeGroundSquirrel	10	2(15)	14	4	4.600000	2	1	1.400000	2	1	1.400000	2	1	1.400000
antpitta	15	0(16)	24	9	9.348148	**0**	**0**	**0.000000**	**0**	**0**	**0.000000**	**0**	**0**	**0.000000**
aquaticSnailGenus	20	22(2)	34	12	12.283333	22	8	8.225000	32	10	10.255000	22	8	8.225000

Figura 16 – Tabla comparativa de distancias Agreement entre los árboles filogenéticos.

Análisis de los resultados obtenidos

Para realizar la comparación entre nuestro método y los métodos de distancia de PAUP para la generación de árboles filogenéticos utilizamos doscientos datasets como base para nuestras pruebas. Estos datasets fueron obtenidos del GenBank [22] donde se almacenan gran parte de los datasets utilizados en las publicaciones relacionadas con la biología. Cada dataset contiene entre quince y veintinueve secuencias de ADN mitocondrial (formato PopSet [23]). Las secuencias de los datasets escogidos no superan las 2000 bases.

Como fue explicado anteriormente, para cada dataset se construyó un árbol filogenético utilizando nuestro método y un árbol para cada una de las distancias de PAUP y los métodos de parsimonia y máxima similitud. El análisis de los resultados consiste en comparar estos árboles utilizando la distancia agreement. Cabe destacar que no se garantiza que los resultados de PAUP sean exactos, ya que estos fueron obtenidos utilizando un método heurístico de construcción de árboles filogenéticos.

La imagen de la distancia agreement es el rango [0, cantidad de secuencias de los datasets]. Como la cantidad de secuencias biológicas varía a lo largo de los dataset, se decidió normalizar esta métrica definiendo la función de similitud *s*:

$$S: arbolBinario \times arbolBinario \rightarrow [0,1]$$
$$S(T,U) = 1 - \frac{d_a(T,U)}{n}$$

Dónde $d_a(T,U)$ es la distancia Agreement entre los árboles *T* y *U*. Y *n* es la cantidad de hojas del árbol (secuencias del dataset).

La imagen de esta función es el intervalo [0, 1], donde el valor uno denota que los dos árboles comparados son idénticos mientras que los valores cercanos a cero representan que los árboles comparados son totalmente distintos.

En el Anexo I se muestra la tabla completa de los resultados obtenidos en las pruebas de nuestro sistema utilizando 200 datasets de secuencias biológicas. Esta tabla se encuentra representada en base a la función de similitud recién definida. A partir de la misma fueron realizados los siguientes análisis de resultados.

El compresor elegido para correr estas pruebas fue el GenCompress debido a fue el compresor que mejor resultados obtuvo aplicado a secuencias pequeñas, como

las que conforman los distintos datasets utilizados (ver sección Análisis de compresores).

Comparación general entre la NCD y las otras métricas de PAUP

En la Figura 17 se resumen los resultados obtenidos por nuestro algoritmo en comparación a los generados con las distintas métricas del método de distancias de PAUP. Los valores del gráfico están expresados en base a la función de similitud agreement aplicada para los árboles filogenéticos generados utilizando nuestro algoritmo y los construidos empleando las métricas de PAUP.

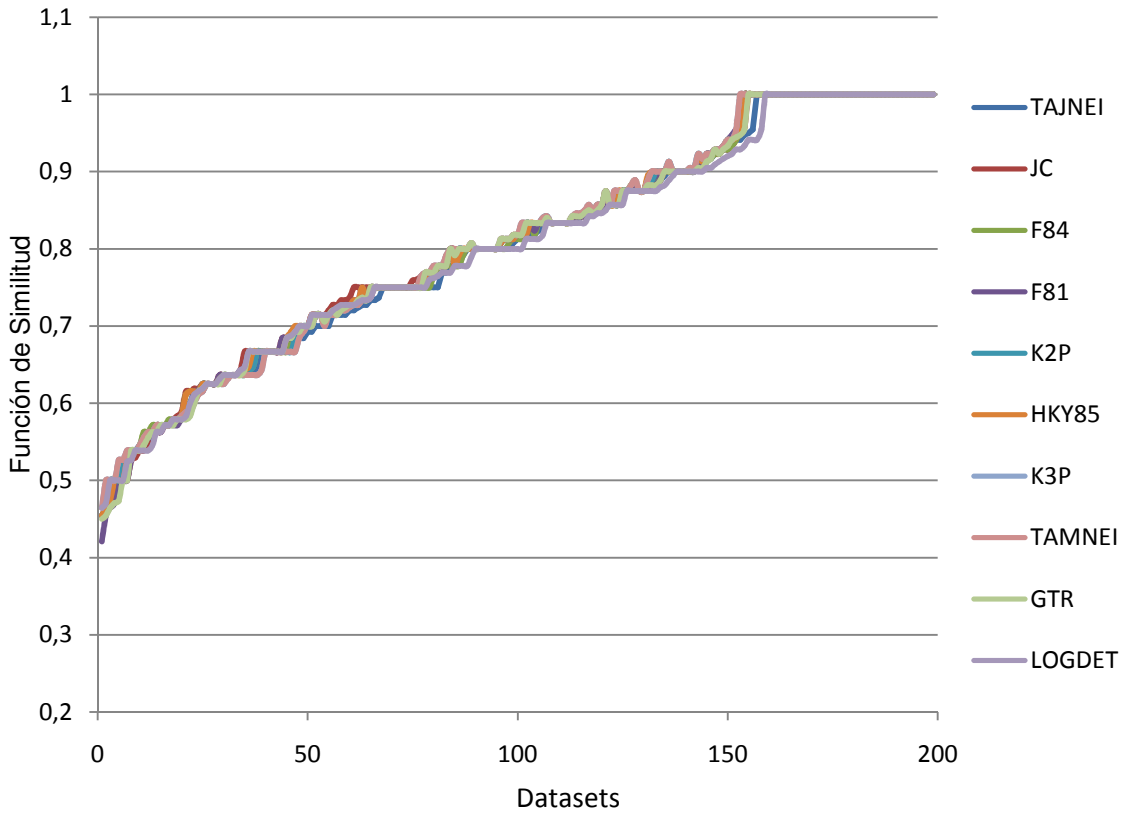


Figura 17– Comparación general entre la NCD y las métricas de PAUP.

El eje X representa los doscientos datasets utilizados en las pruebas. En cambio el eje Y muestra el valor de la función de similitud aplicada al árbol filogenético construido por nuestro método junto al árbol filogenético generado utilizando PAUP. Este valor es calculado para los árboles construidos por cada una de las métricas provistas por PAUP (TAJNEI, JC, F84, etc.).

Formalmente podemos representar la función aplicada en este gráfico de la siguiente manera:

$$f: dataset \times metrica \rightarrow [0,1]$$
$$f(d,m) = S(arbol_{NCD}(d), arbol_{PAUP}(d,m))$$

Donde:

- La función $arbol_{NCD}(d)$ devuelve el árbol filogenético construido utilizando nuestro método para el dataset *d*.
- La función arbolPAUP(d,m) devuelve el árbol filogenético construido con PAUP utilizando la métrica m para el dataset d.
- La función S(árbol, árbol) devuelve el valor de similitud agreement para los dos árboles filogenéticos recibidos como parámetro.

La función *f* está basada en el uso de la función de similitud agreement *S*, por lo tanto cuando el valor de la función *f* se devuelve valores cercanos al uno representa que los árboles comparados son similares. En cambio, cuando el valor de la función es cercano al cero significa que los árboles comparados son muy distintos.

Cabe destacar que el objetivo de este gráfico es el de poder visualizar los resultados de nuestro método en comparación con los resultados del método de distancias de PAUP aplicado a todas sus métricas para todos los datasets en general. Es decir, no se intenta distinguir la diferencia entre el resultado de cada dataset en particular aplicado a todas las métricas de PAUP. Por lo tanto, los valores de la función de similitud fueron ordenados de menor a mayor. Esto implica que el dataset utilizado en cada posición del eje X no sea el mismo para cada una de las métricas de PAUP. Este tipo de representación ayuda a visualizar la comparación general de los métodos.

El grafico de la Figura 18 muestra un histograma de los desvíos estándar entre los resultados de la similitud agreement por cada uno de los datasets utilizados.

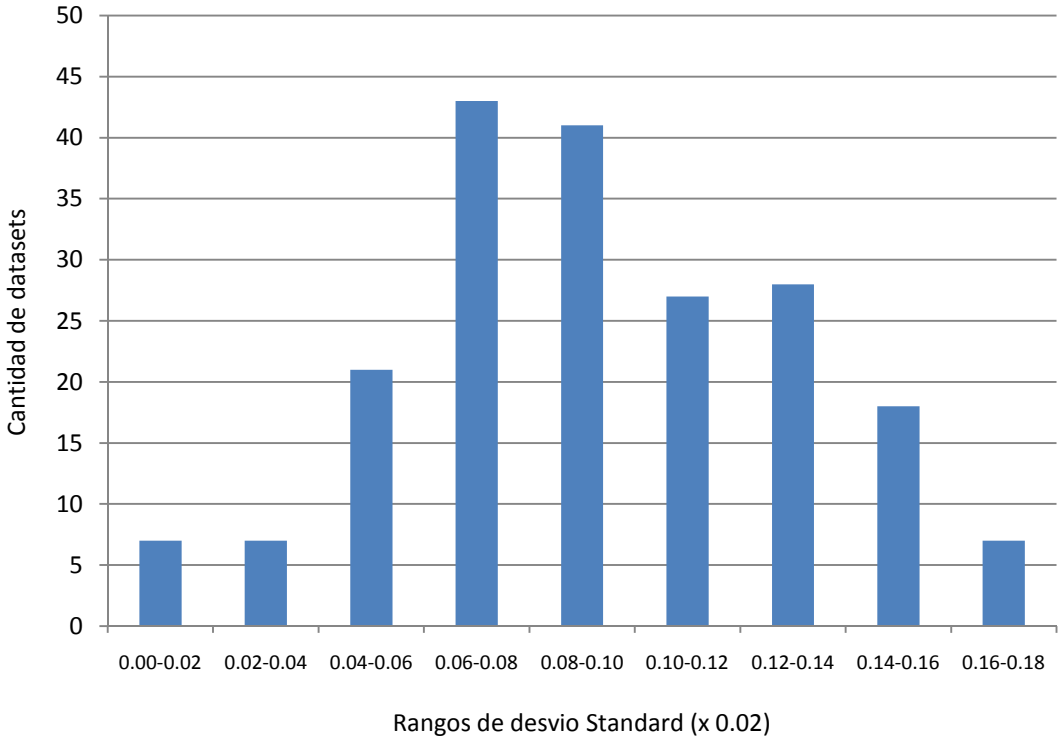


Figura 18 - Desvío estandar de la similitud agreement

En cuanto a los resultados de los gráficos anteriores, podemos observar lo siguiente:

- El menor valor de similitud Agreement para todas las métricas de PAUP está en el rango (0.4 , 0.5)
- Para prácticamente todas las métricas de PAUP se obtuvieron casi un 25 % de datasets para los cuales los árboles filogenéticos construidos por nuestro método fueron indistinguibles en comparación a los construidos con el método de distancia de PAUP.
- A grandes rasgos por lo menos el 50% de los resultados arrojados por nuestro método cuentan con una similitud mayor a 0.8 en comparación al método de distancias de PAUP.
- El resultado de la función *f* sigue la misma curva para todas las métricas de PAUP.

- El análisis por cada datasets, de la comparación de la métrica NCD con respecto al resto de las métricas disponibles, refleja un comportamiento homogéneo en cuanto a la medida agreement. El desvío estándar de cada una de las muestras es en promedio de 0,07.

Este mismo análisis fue realizado comparando nuestro algoritmo con los métodos de Máxima Similitud y Parsimonia.

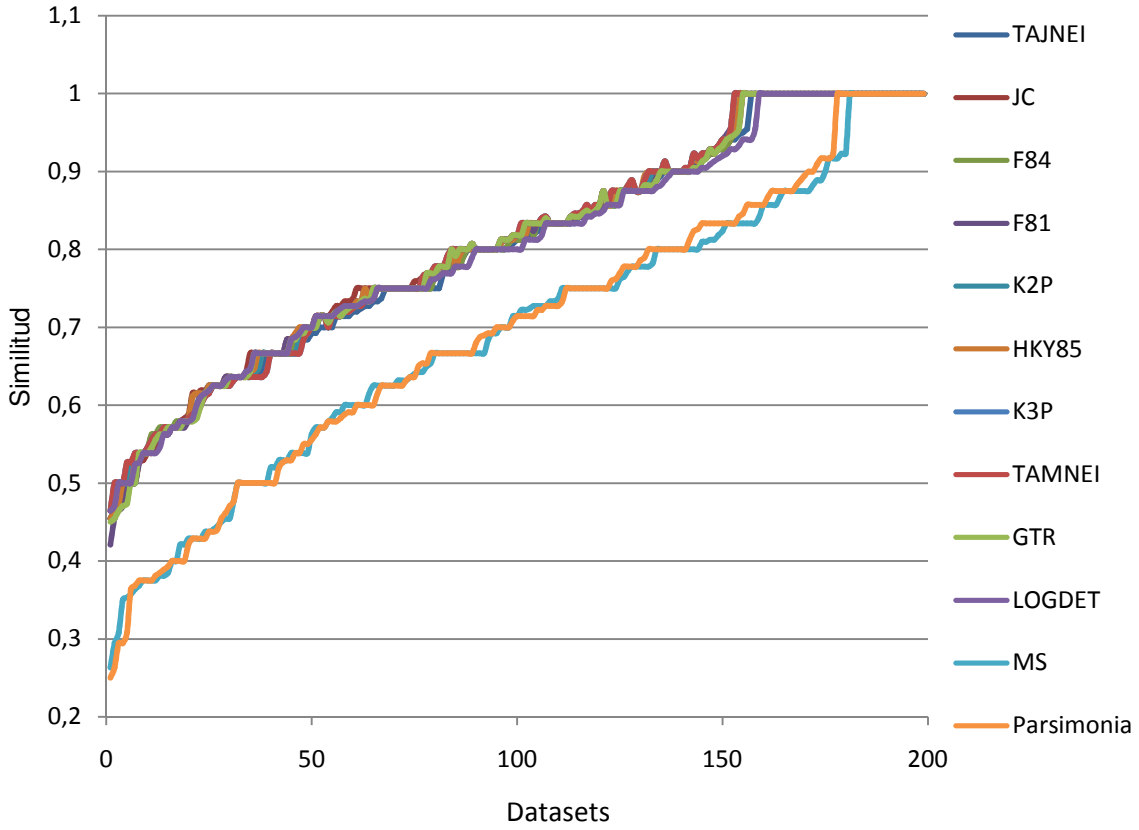


Figura 19 –Comparación general entre la NCDy las métricas de PAUP incluyendo Máxima Similitud y Parsimonia

El gráfico de la Figura 19 muestra claramente que nuestro algoritmo se asemeja mucho más al método de distancia de PAUP que a lo métodos de máxima similitud y parsimonia. Algunos puntos que avalan esta conclusión son:

- El menor valor de similitud Agreement para los métodos de Máxima Similitud y Parsimonia se encuentran en el rango (0.2 , 0.3) a diferencia del (0.4 , 0.5) del método de distancias.

- Tan solo un 10% de los árboles filogenéticos basados en NCD fueron indistinguibles a los generados utilizando el método de máxima similitud y parsimonia.

- 33% de los resultados arrojados por nuestro método cuentan con una similitud mayor a 0.8 en comparación a los métodos de máxima similitud y parsimonia.

Comparación de árboles idénticos

El histograma de la Figura 20 muestra la cantidad de datasets para los cuales el árbol filogenético obtenido por nuestro algoritmo fue similar al elaborado utilizando cada una de las métricas de PAUP.

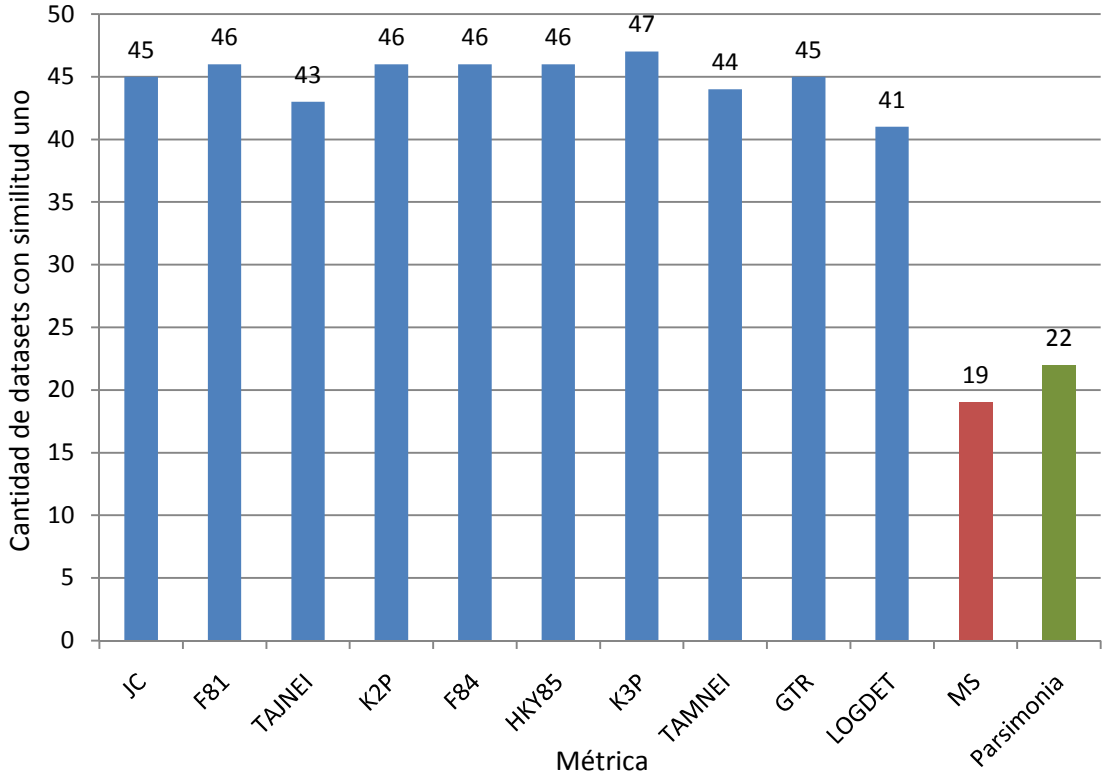


Figura 20 — Comparación de árboles idénticos

En el gráfico podemos notar que la K3P es la métrica del método de distancia de PAUP que mas se asemeja a nuestro algoritmo cuando se comparan la cantidad de datasets para los cuales el árbol filogenético obtenido por ambos algoritmos es idéntico. Por el contrario la métrica LOGDET es la que peor resultados arrojó ya que

solo 41 árboles resultaron idénticos luego de la ejecución de ambos métodos. Por otro lado podemos ver que para la mayoria del resto de las métricas del método de distancias de PAUP se obtuvieron entre 45 y 46 árboles filogenéticos idénticos. Por último tan solo 19 árboles resultaron idénticos al comparar nuetro algoritmo con el método de Máxima Similitud y 22 con el método de Parsimonia.

Sobre el total de 200 datasets, entre el 20% y 25% tuvo un comportamiento idéntico con respecto al método de distancia de Paup. Este valor se reduce a un 10% con respecto al método de maxima similitud.

En ningún momento se hizo distinción o mención, a las posibles interpretaciones de las distancias utilizadas, de las matrices calculadas o de la naturaleza de los datasets. Esto significa, que en estos casos, la métrica NCD tuvo un comportamiento sintácticamente igual con respecto a las otras métricas.

Función de similitud y cantidad de secuencias del dataset

El histograma de la Figura 21 muestra el promedio de la función de similitud obtenida teniendo en cuenta la cantidad de secuencias que contiene cada dataset. Este análisis fue realizado para cada una de las métricas de PAUP. A partir de este gráfico se desea determinar si la cantidad de secuencias que contenga cada dataset influye en los resultados de nuestro algoritmo.

En el eje X del gráfico se detalla la cantidad de secuencias que describe cada columna del histograma. Entre paréntesis se detalla la cantidad de datasets que contengan dicha cantidad de secuencias, esta información es vital para poder obtener conclusiones acerca de la validez de los resultados.

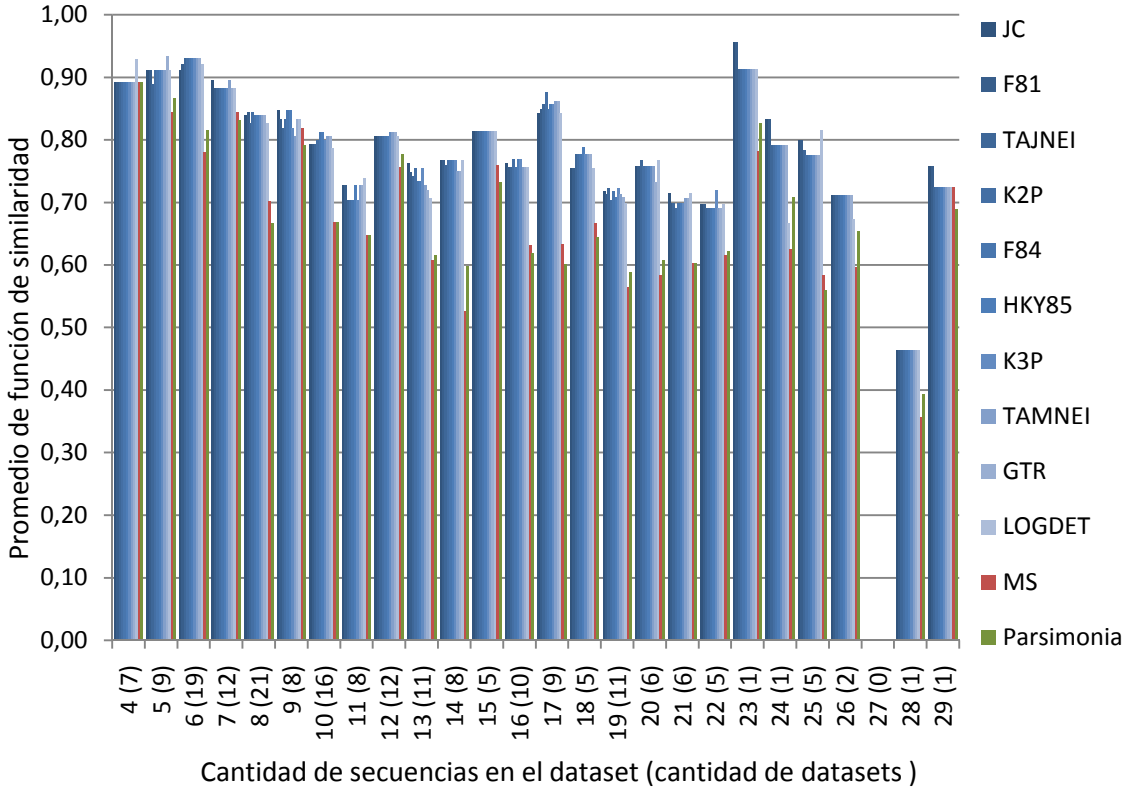


Figura 21 – Comparación del promedio de la función de similitud según la cantidad de secuencias de los datasets.

En este gráfico podemos notar que los datasets que contienen entre cuatro y seis secuencias tienen un promedio de función de similitud cercano a 0.9. En cambio para los dataset cuya cantidad de secuencias es mayor o igual a siete el promedio de la función de similitud se encuentra en el rango de (0.7, 0.85). En este caso no se nota una reducción de los resultados de la función de similitud a partir del incremento en la cantidad de secuencias del dataset.

En el gráfico se destacan los datasets con veintitrés y veintiocho secuencias biológicas. Éstos tienen un promedio de similitud de 0.9 y 0.45 respectivamente. Estos resultados extraordinarios se explican teniendo en cuenta que solamente existe un dataset con esa cantidad de secuencias para cada caso. Por lo tanto estos los podemos considerar outliers dejándolos fuera del análisis.

En todos los casos el promedio de la función de similitud para el método de máxima similitud y parsimonia es menor en comparación con las distintas métricas de método de distancias de PAUP.

A partir de este resultado podemos concluir que la cantidad de secuencias que contenga el dataset no influye de manera notable en los resultados del algoritmo para datasets cuya cantidad de secuencias sea superior a siete.

Conclusiones

En este trabajo se estudió la utilización de la medida de Similitud Universal definida por Vitányi en un problema importante de biología como lo es la inferencia de árboles filogenéticos.

Dentro de este marco, primero desarrollamos un estudio sobre algunos compresores existentes para determinar cuál era el más apropiado para aproximar la NCD. Gracias a ese estudio determinamos que el GenCompress es el que mejor cumple las propiedades de compresor normal propuestas por Vitányi en [5] para archivos de longitud menor a 8Mb mientras que para archivos mayores a 8Mb, el LRZip fue el que obtuvo mejores resultados. Los datasets utilizados en las distintas publicaciones relacionadas con este trabajo, que aportaron información al mismo y figuran en sus referencias, no exceden los 8Mb por secuencia.

Finalmente, luego del estudio de los compresores, nos avocamos al estudio de la medida de NCD aplicada a la filogenia. Dentro de este análisis podemos remarcar los siguientes aspectos:

- Los métodos computacionales tradicionales utilizados para generar árboles filogenéticos se pueden clasificar, a grandes rasgos, en dos grupos. Por un lado, se encuentran los métodos que resuelven el problema de manera sintáctica, sin utilizar conocimiento del dominio, y por otro aquellos que necesitan de un modelo para su aplicación. El método estudiado en este trabajo se encuentra dentro del primer grupo dado que es un método de clasificación general [24]. La utilización de la métrica NCD, permite contar con un método basado en distancias que no requiere el alineamiento de las secuencias a clasificar. Este hecho es una ventaja dado que no se necesita ningún preprocesamiento de las secuencias biológicas a estudiar. El resto de los métodos mencionados en este trabajo necesitan alinear las secuencias como paso previo a su aplicación.

- Existen tres grandes familias de métodos para la inferencia de árboles filogenéticos: los métodos de distancia, los métodos de parsimonia y los de máxima verosimilitud. El primero, está basado básicamente en el análisis y comparación sintáctica de las distintas cadenas. Los dos siguientes, se basan en modelos e interpretaciones, fundamentadas desde la biología. Son métodos en donde la semántica es mucho más predominante que la sintaxis. Definitivamente, la utilización de la NCD para este tipo de problemas, cae en la taxonomía de los primeros (al igual que los distintos métodos provistos por PAUP utilizados en este

trabajo). Sin embargo, en la actualidad los método predominantes para el cálculo de árboles filogenéticos son los métodos de parsimonia y máxima verosimilitud.

- Si bien los resultados obtenidos en este trabajo muestran que el 50% de los árboles filogenéticos generados con la técnica de Vitányi cuentan con una similitud mayor al 80% con los árboles obtenidos con los métodos de distancia de PAUP, y que el 75% cuentan con una similitud mayor a 75%, no podemos argumentar que esta técnica sea aplicable para la generación de árboles filogenéticos debido a que los métodos de distancias de PAUP no son los utilizados en la actualidad por la comunidad científica. De todas maneras, en comparación con los método predominantes actualmente (parsimonia y máxima verosimilitud), este método tiene un comportamiento similar a los otros métodos de distancia. Esto abre las puertas a una posible aplicación de la NCD al estudio filogenético: puede ser utilizado como generador de árboles iníciales para los métodos filogenéticos exhaustivos como el método de parsimonia y máxima verosimilitud o para algunos métodos que utilicen algoritmos de búsqueda local.

- Todos los métodos estudiados en este trabajo requieren cierta supervisión experta, pues asumen ciertas propiedades, hipótesis o modelos para poder evaluar la similitud entre un conjunto de secuencias. Por su parte, la NCD no requiere ninguna supervisión, ni posee fundamentación biológica alguna para su aplicación. De todas maneras, en todos los casos, los resultados y su interpretación sí deben ser supervisados.

- La mayoría de las publicaciones mencionadas en este trabajo sobre la clasificación de secuencias biológicas utilizando el método propuesto por Vitányi no fundamentan la elección del compresor ni cotejan si el compresor escogido cumplen con las propiedades establecidas por Vitányi [5]. Tampoco mencionan o discriminan en sus análisis, contra qué tipo de métodos establecen sus comparaciones.

Trabajos Futuros

En esta sección se presentarán líneas de trabajo futuro con respecto a nuestro trabajo, incluyendo ampliaciones al mismo, así también como análisis para desarrollar posibles mejoras.

- **Sobre grandes volúmenes de datos**

  La aplicación desarrollada en este trabajo fue probada utilizando doscientos datasets obtenidos del sitio Web de GenBank. Cada uno de estos dataset contiene entre quince y veintinueve secuencias de ADN mitocondrial y la longitud de cada secuencia de ADN no supera las 2000 bases. Esto implica que la longitud de cada archivo utilizado es menor a 2 KB.

  Algunos problemas bioinformáticas requerirán la manipulación de archivos considerablemente más grandes. No es trivial ampliar este trabajo para que contemple grandes volúmenes de datos. En muchos casos, requerirá sin duda abandonar la utilización de compresores estándares y abordar el desarrollo de un compresor ad-hoc.

- **Desarrollo de un compresor**

  La utilización del GenCompress como compresor limita el tamaño de los archivos a clasificar debido a que el orden de complejidad del mismo es demasiado alto. Esto implica que los archivos cuyo tamaño ronden en el orden de los cientos Mb no sean posibles de comprimir. Esto es de suma importancia en ciertos dominios, donde los archivos con secuencias de ADN son de gran tamaño. En el desarrollo de este nuevo compresor, no sólo se debe considerar la eficiencia (tanto en tiempo como en uso de memoria), sino también que cumpla las propiedades con las que Vitanyi caracterizó a los compresores normales.

- **Aplicación en otros problemas bioinformáticos**

  En [23] la NCD definida por Vitányi fue utilizada en la clasificación automática de documentos y, en este trabajo, fue aplicada en la inferencia de árboles filogenéticos. Por ende, una nueva línea de investigación posible que se desprende de este trabajo es la utilización de esta métrica para la clasificación de otros tipos de secuencias biológicas tales como proteínas. En pruebas preliminares para el desarrollo de este trabajo, la aplicación de la NCD

clasificó correctamente un dataset de 5 cadenas betas de hemoglobina humana. El dataset y la clasificación se pueden encontrar en la sección 2.4.3 del trabajo [13].

Para obtener una medida de similitud entre secuencias, la NCD se aprovecha de la información en común presente entre las mismas. La obtención y análisis de esta información también abre las puertas para nuevos trabajos: el estudio de los patrones comunes entre secuencias, con el fin de inferir no sólo presencia en distintos organismos, presencia en distintas regiones del ADN o dar soporte a otros estudios biológicos y/o bioinformáticos.

- **Otros dominios de problemas**

  Cualquier dominio de problema en donde los objetos del dominio puedan ser representados como secuencias de caracteres, es plausibles de someterse a la NCD. En cada caso, se deberá analizar y determinar que compresor utilizar.

Anexo I

En la siguiente tabla se muestran las distancias de nuestro método con respecto a las principales distancias de PAUP para cada uno de los 200 datasets utilizados en nuestras pruebas.

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
Aethalops	21	0.67	0.62	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.52	0.38	0.43
AfricanHominoids	12	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.58	0.58
AfricanTilapiineFishes	12	0.83	0.75	0.75	0.75	0.75	0.75	0.75	0.83	0.83	0.83	0.75	0.67	0.67
AmericanCollaredLemmings	17	0.82	0.76	0.82	0.82	0.76	0.76	0.76	0.76	0.82	0.88	0.76	0.53	0.47
AncientCetaceanlineages	9	0.67	0.67	0.56	0.56	0.67	0.78	0.78	0.56	0.67	0.67	0.67	0.67	0.67
AncientCetaceanlineages2	6	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
AncientCetaceanlineages3	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	1.00
angiosperms	13	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.62
angiosperms2	17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.88
anopheles-arabiensis	18	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.78	0.72	0.72
AnophelesOswaldoi	25	0.84	0.84	0.84	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.88	0.64	0.68
AnophelesPunctulatus	11	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.73	0.73	0.73
anophelinae2	19	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.58	0.63	0.53
AntelopeGroundSquirrel	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.80	0.50	0.50
antpitta	15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.87	0.87
aquaticSnailGenus	20	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.45	0.45	0.50	0.45	0.55
AustralianPaintedSnipe	7	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.57	0.57
AustralianPaintedSnipe2	7	0.71	0.86	0.71	0.71	0.71	0.71	0.71	0.71	0.86	0.71	0.71	0.43	0.43
AvesCettiidae	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	0.83
AvianGenusLophura	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.75	0.88
AvianGenusLophura2	4	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
avianGenusPipilo	9	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
BaicalianEndemicSergentia	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
BaileysPocketMouse	15	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.60
batMyotisMyotis	16	0.88	0.75	0.75	0.81	0.81	0.75	0.75	0.81	0.75	0.75	0.81	0.50	0.50

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
bats	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
bears	7	0.86	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
birchLeafminingSawflies	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50
blowflies	15	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.73	0.80
BodiedSageGrouse	22	0.73	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.59	0.59
BosTaurus	19	0.68	0.68	0.68	0.68	0.63	0.63	0.63	0.63	0.68	0.68	0.68	0.42	0.58
brachypterousGrasshopper	11	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.45	0.45
BrachyramphusMurrelets	5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
BrownSeaweeds	10	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
BrownSeaweeds2	18	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.56
BrownSeaweeds3	22	0.59	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.59	0.59
bullTrout	8	1.00	1.00	1.00	0.88	1.00	1.00	1.00	1.00	0.88	0.88	0.75	0.88	0.88
BushWarbler	12	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.92	0.83
Calliphoridae	8	0.88	0.75	0.75	0.63	0.75	0.63	0.63	0.75	0.63	0.63	0.75	0.63	0.63
catfish	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.75	0.75
Catostomidae	19	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
CaucasianSalamander	5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00
CaucasianWoodMice	19	0.74	0.68	0.68	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.63	0.58	0.58
CaveSalamander	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.60	0.60
Cedrus	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.67
Cedrus2	6	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
Cedrus3	6	1.00	0.83	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.83
Cedrus4	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	0.83
Cedrus5	6	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.67	0.67
Cedrus6	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Cervinae	21	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.76	0.81	0.76
ChinaSeasGroupers	5	1.00	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
ChineseGiantSalamanders	17	0.82	0.88	0.88	0.94	0.94	0.88	0.94	0.94	0.94	0.94	0.94	0.53	0.53
ChlamydiaTrachomatis	16	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.38	0.38
cotingas	25	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.76	0.80	0.52
CraxViridirostrisSclater	12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.75

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
Curassows	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.80	0.70
Cycleptus	20	0.90	0.80	0.80	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.90	0.60	0.55
CytochromeRodents	5	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
DecapterusMacrosoma	13	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.54	0.38
DiplodactylusStenodactylus	12	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
DominicanAnole	4	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
DrosophilaObscura	22	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.77	0.73
DrosophilaSimulans	8	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63
DrosophilaSimulans10	8	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.38	0.38
DrosophilaSimulans11	8	0.88	0.75	0.75	0.75	0.88	0.75	0.75	0.88	0.88	0.88	0.88	0.63	0.63
DrosophilaSimulans12	8	0.75	0.63	0.75	0.75	0.75	0.75	0.63	0.63	0.75	0.75	0.75	0.63	0.63
DrosophilaSimulans2	8	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.63
DrosophilaSimulans3	8	0.88	0.88	1.00	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
DrosophilaSimulans4	8	0.63	0.75	0.63	0.63	0.63	0.63	0.75	0.63	0.75	0.75	0.75	0.38	0.38
DrosophilaSimulans5	8	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.38	0.38
DrosophilaSimulans6	8	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.50	0.50
DrosophilaSimulans7	9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.78	0.67
DrosophilaSimulans8	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.75
DrosophilaSimulans9	8	0.75	0.63	0.63	0.63	0.63	0.75	0.75	0.63	0.63	0.63	0.63	0.63	0.63
dungBeetles	11	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
EndangeredEuropeanMink	10	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.60	0.60
Entomopathogenic	13	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.77	0.77	0.62	0.77
Entomopathogenic2	12	0.75	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.50	0.50
euphausiid	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	1.00
figWasps	14	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.43	0.79
forcipulataceanSeaStars	25	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.44	0.44
FreshwaterBarbs	13	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.92	0.92
FreshwaterFish	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
freshwaterFishSpecies	13	0.54	0.62	0.62	0.62	0.54	0.62	0.62	0.54	0.54	0.54	0.54	0.31	0.31
FreshwaterSardines	5	0.80	1.00	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.60	0.60
FrogsScinax	24	0.88	0.83	0.83	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.67	0.63	0.71

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
Funariidae	18	0.89	0.83	0.83	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.83	0.72	0.83
galagos	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.80	0.80
galagos2	10	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.60	0.70
galagos3	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.70	0.80
gardenLizards	21	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.81	0.86
Gastropoda	22	0.64	0.45	0.45	0.45	0.45	0.45	0.45	0.64	0.45	0.45	0.50	0.45	0.50
GastropodFrigidoalvania	12	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.75	0.75
GoatsBreeds	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.80	0.90
gophers	10	1.00	0.80	0.70	0.70	0.80	0.80	0.80	0.80	0.80	0.80	0.70	0.60	0.60
GreatApes	5	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
GreatApes2	14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.64	0.71
greyParrot	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.86
guineaPigs	17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94	0.76	0.59
HarpSeals	13	0.77	0.77	0.69	0.69	0.77	0.62	0.62	0.77	0.69	0.69	0.54	0.38	0.46
heron	16	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.81	0.75
HesperotettixViridis	19	0.63	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.37	0.37
holarcticRodent	9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.78	0.78
HumanBackflow	8	1.00	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.75
Hylobates	25	0.88	0.88	0.88	0.88	0.84	0.84	0.84	0.84	0.84	0.84	0.92	0.52	0.76
Hypocrea	10	0.90	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.90	0.80
HypocreaPulvinata	15	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
Isoptera	17	0.65	0.59	0.65	0.59	0.65	0.65	0.65	0.59	0.65	0.71	0.65	0.29	0.29
Isoptera2	17	0.88	0.88	0.88	0.88	1.00	0.88	0.88	0.88	0.94	0.88	0.88	0.53	0.53
Ivindomyrus	11	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.73	0.36	0.36
jacanas	8	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.25
LacertaViridis	14	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.64	0.86
LagopusMutus	25	0.76	0.76	0.76	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.52	0.40
lionfish	28	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.36	0.39
LowlandTropicalSalamanders	14	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.50
LycaenaThersamon	6	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
MadagascarTortoises	7	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
malariaVectors	12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
marshRabbit	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	1.00	1.00
Marsupialia	8	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.75	0.75
marsupials	9	0.67	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.56	0.78	0.78	0.89	0.78
MediterraneanLizard	21	0.57	0.62	0.57	0.57	0.57	0.57	0.57	0.57	0.62	0.62	0.62	0.38	0.38
Megascolecidae	13	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
Melanesian	16	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.75	0.81
Melanoseps	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.88
Merluccius	26	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.73	0.65	0.65
Microtus	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
MicrotusBavaricus	12	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
molecularClock	14	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.57	0.57
moleRats	12	0.83	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.83
multigenePlastid	26	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.54	0.65
MuntjacDeer	13	0.92	0.92	0.85	0.77	0.92	0.77	0.77	0.92	0.69	0.77	0.77	0.92	0.85
MyadestesSPP	7	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
NearcticPikas	5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
NeisseriaMeningitidis	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
NeomysWaterShrews	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.88
NeotropicalFrogs	11	0.82	0.82	0.82	0.64	0.64	0.64	0.82	0.64	0.82	0.82	0.82	0.82	0.64
NeotropicalRodent	17	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.88	0.88	0.94	0.82	0.82
NorthAmericanChthamalus	11	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.91	0.73	0.91
NorthernEuropeanLynx	4	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	1.00	0.75	0.75
Odontiphoridae	16	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
OrderAnaspidea	16	0.69	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.69	0.75	0.75
OrderAnaspidea2	16	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.69
Palaearctic	5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Parabotia	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.86	0.71
ParadoxicalFrogs	17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
percina-macrocephala	18	0.94	0.89	0.89	0.89	0.89	0.89	0.94	0.89	0.94	0.94	0.89	0.78	0.72
Pleistocene	16	0.81	0.88	0.81	0.75	0.88	0.81	0.81	0.88	0.81	0.81	0.75	0.44	0.44

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
PleistoceneRefugia	13	0.77	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.69	0.69
PoisonFrogs	11	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
PoisonFrogs3	11	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64
Procyonidae	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Protobothrops	19	0.89	0.84	0.89	0.89	0.84	0.89	0.89	1.00	0.89	0.84	0.79	0.63	0.63
Pseudocolaptes	7	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
PteroisMiles	13	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.54	0.54
PygmySunfishes	9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
rabbitfishes	23	0.96	0.96	0.96	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.78	0.83
Rafflesiaceae	22	0.68	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.50	0.50
RainforestFrogs	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ravens	6	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.67	0.67
ravens2	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	0.83
Raymunida	21	0.90	0.90	0.90	0.90	0.86	0.90	0.90	0.90	0.90	0.90	0.90	0.76	0.71
RedBackedVoles	18	0.56	0.56	0.56	0.61	0.61	0.61	0.61	0.61	0.56	0.56	0.61	0.44	0.39
RedFox	14	0.57	0.71	0.71	0.64	0.71	0.71	0.71	0.71	0.57	0.57	0.71	0.50	0.50
RockfishSubgenusSebastomus	20	0.90	0.90	0.90	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.80	0.70	0.70
rockLobsters	9	0.78	0.78	0.78	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.89	0.89
rockLobsters2	9	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
rodentMalaria	16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.88
rodents	20	0.85	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.75	0.35	0.40
Rupicapra	16	0.81	0.75	0.75	0.75	0.75	0.75	0.88	0.75	0.75	0.75	0.81	0.44	0.44
SalmoTrutta	10	0.70	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.40	0.40
SalmoTrutta2	10	0.70	0.80	0.80	0.80	0.80	0.90	0.90	0.80	0.90	0.90	0.80	0.50	0.50
scombrids	6	0.83	0.67	0.67	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.67	0.67
Seabreams	10	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
Serrasalmidae	19	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.74	0.89
shrimp	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SigmodonHispidus	12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.92	0.92
SigmodonOchrognathus	5	1.00	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	1.00	0.80	0.80	0.80
SileneVulgaris	14	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.43	0.43

Datasets	#Taxas	P	JC	F81	TAJNEI	K2P	F84	HKY85	K3P	TAMNEI	GTR	LOGDET	MS	PARSI
SileneVulgaris2	14	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.43	0.43
sisoridCatfishes	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.86	1.00
SouthAmericanPinnipeds	19	0.53	0.47	0.42	0.47	0.53	0.58	0.47	0.53	0.47	0.47	0.53	0.26	0.26
SpiderMacrotheleCalpeiana	19	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.63	0.58
SpiderMacrotheleCalpeiana2	19	0.95	1.00	0.95	0.95	0.95	1.00	1.00	1.00	0.95	0.95	0.89	0.68	0.79
spiny-headed-treefrog	15	0.80	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.80	0.73
StreambedMicrostructure	7	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	1.00	1.00
sturgeon	19	0.58	0.53	0.53	0.68	0.53	0.53	0.53	0.53	0.58	0.58	0.58	0.42	0.42
SwallowPopulations	13	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.69	0.69
Sympatry	6	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.67
Sympatry2	10	0.70	0.50	0.60	0.60	0.60	0.70	0.70	0.60	0.60	0.60	0.60	0.40	0.40
TegillarcaGranosa	17	0.59	0.53	0.47	0.53	0.59	0.53	0.53	0.59	0.53	0.47	0.47	0.35	0.29
Thomasomys	12	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.92
torrentSalamander	6	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	1.00
TribeMolopini	10	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	1.00	0.90
Tupinambis	20	1.00	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.75	0.80
Verticillium	29	0.83	0.76	0.76	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.69
Vespertilionidae	21	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.48	0.48
Vireos	20	0.95	0.80	0.80	0.95	0.90	0.90	0.90	0.90	0.95	0.80	0.80	0.65	0.65

Referencias

[1] C.H. Bennett, P. Gacs, M. Li, P.M.B. Vitányi, and W. Zurek. Information Distance, IEEE Transactions on Information Theory, 44:4(1998), 1407–1423.

[2] R. Cilibrasi, P.M.B. Vitányi, Clustering by compression, IEEE Transactions on information theory, vol. 51, no 4, Abril 2005, 1523–1545.

[3] A.N. Kolmogorov, Three approaches to the definition of the concept 'quantity of information', Problems in Information Transmission, 1(1):1-7, 1965.

[4] G. Chaitin. A theory of program size formally identical to information theory. J. Assoc. Comput. Mach., 22:329-340, 1975.

[5] M. Li, X. Chen, X. Li, B. Ma, P.M.B. Vitányi. The similarity metric, Proc. 14th ACM-SIAM Symposium on Discrete Algorithms, 2003.

[6] W. Goddarg, E. Kubicka, G. Kubicki, F. McMorris. The Agreement Metric for Labeled Binary Trees, 1994.

[7] Evolutionary tree from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368-376

[8] Ziv, J. y Lempel, A., A universal algorithm for sequential data compression. IEEE Transactions on Information Theory. 23:337-343, 1977.

[9] Ziv, J. y Lempel, A., Compression of individual sequences via variable rate coding. IEEE Transactions on Information Theory. 24:530-536, 1978.

[10] Burrows, M. y Wheeler, D., A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994.

[11] Grumbach, S. y Tahi, F., A new Challenger for compression algorithms: genetic sequences, J. Information Processing and Management, 30(6):866-875, 1994.

[12] X. Chen, S. Kwong, y M. Li, A compression algorithm for DNA sequences and its applications in genome comparison. RECOMB, page 107, 2000.

[13] S. Vinga. J. Biological sequence analysis by vector maps: alignment-free comparison of DNA and proteins. Tesis de Doctorado. Universidade Nova de Lisboa. 2005.

[14] Sitio web del compresor LRZIP. http://ck.kolivas.org/apps/lrzip/

[15] Sitio web del compresor LZMA: http://www.7-zip.org/

[16] Sitio web del compresor LZO: http://www.oberhumer.com/opensource/lzo/

[17] Ensembl FTP Site. http://www.ensembl.org/info/downloads/ftp_site.html

[18] Algoritmo Neighbor-Joining. http://mbe.oxfordjournals.org/cgi/reprint/4/4/406

[19] Classification of oligonucleotide fingerprints: application for microbial community and gene expression analyses. http://bioinformatics.oxfordjournals.org/cgi/reprint/21/14/3122.

[20] UPGMA: Sneath & Sokal 1973. Numerical Taxonomy.W.H. Freeman and Company, San Francisco, pp 230-234

[21] Feng, D. F. y Doolittle, R. F. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol., 25, 351–360.

[22] GenBank: www.ncbi.nlm.nih.gov

[23] PopSet Home: http://www.ncbi.nlm.nih.gov/sites/entrez?db=popset

[24] Capparelli A y Urtasun M. Clasificación automática de textos basada en la métrica de similitud universal de Vitanyi. Tesis de Licenciatura. Universidad de Buenos Aires. 2005.

[25] Needleman, Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins, J Mol Biol. **48**(3):443-53.

[26] Smith TF, Waterman MS. 1981. "Identification of Common Molecular Subsequences".

[27] PHYLIP: http://evolution.genetics.washington.edu/phylip.html

[28] PAUP: http://paup.csit.fsu.edu/

[29] TNT: http://www.zmuc.dk/public/Phylogeny/TNT/

[30] BLAST: http://blast.ncbi.nlm.nih.gov

[31] Kocsor A, Kertész-Farkas A, Kaján L, Pongor S. Application of compression-based distance measures to protein sequence classification: a methodological study. Bioinformatics. 2006 Feb 15;22(4):407-12. Epub 2005 Nov 29.

[32] N. Krasnogor, D. A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. Bioinformatics, 20, 1015-1021.

[33] Huffman, D. A. - A Method for the Construction of Minimum-Redundancy Codes.

[34] Jukes TH, Cantor CR: Evolution of protein molecules. In Mammalian Protein Metabolism Edited by: Munro H N. New York, Academic Press; 1969:21-132.

[35] Felsenstein J: Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 1981, 17(6):368-376.

[36] Tajima F, Nei M: Estimation of evolutionary distance between nucleotide sequences. Mol Biol Evol 1984, 1(3):269-285.

[37] Kimura M: A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J Mol Evol 1980, 16(2):111-120.

[38] Felsenstein J: Distance methods for inferring phylogenies – a justification. Evolution 1984, 38(1):16-24.

[39] Hasegawa M, Kishino H, Yano T: Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol 1985, 22:160-174.

[40] Kimura M: Estimation of evolutionary distances between homologous nucleotide sequences. Proc Natl Acad Sci U S A 1981, 78(1):454-458.

[41] Tamura K, Nei M: Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. Mol Biol Evol 1993, 10(3):512-526.

[42] Lanave C, Preparata G, Saccone C, Serio G: A new method for calculating evolutionary substitution rates. J Mol Evol 1984, 20(1):86-93.

[43] Rodriguez F, Oliver JL, Marin A, Medina JR: The general stochastic model of nucleotide substitution. J Theor Biol 1990, 142(4):485-501.

[44] Steel MA: Recovering a tree from the leaf colourations it generates under a Markov model. App Math Lett 1994, 7(2):19-24.

[45] Lockhart PJ, Steel MA, Penny D, Hendy MD: Recovering evolutionary trees under a more realistic model of sequence evolution. Mol Biol Evol 1994, 11(4):605-612.