ALGORITMO PARA ANÁLISIS FILOGENÉTICO: UPGMA

SANDRA MILENA ACERO BARRAZA

UNIVERSIDAD DEL NORTE
DIVISIÓN DE INGENIERÍAS
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
BARRANQUILLA
2007

ALGORITMO PARA ANÁLISIS FILOGENÉTICO: UPGMA

SANDRA MILENA ACERO BARRAZA

Tesis propuesta como cumplimiento de los
requisitos para optar al titulo de:

Ingeniera de sistemas

Director:
EDUARDO ENRIQUE ZUREK VARELA PhD.
Ingeniero de Sistemas

UNIVERSIDAD DEL NORTE
DIVISIÓN DE INGENIERÍAS
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
BARRANQUILLA
2007

ALGORITMO PARA ANÁLISIS FILOGENÉTICO: UPGMA

Por

Sandra Milena Acero Barraza

Tesis propuesta como cumplimiento de los
requisitos para optar al titulo de:

Ingeniera de sistemas

Director:
EDUARDO ENRIQUE ZUREK VARELA PhD.
Ingeniero de Sistemas

VoBo_____

Universidad del Norte
2007

Aprobada por _____

_____

_____

_____

Programa autorizado para obtener el titulo

_____

Fecha _____

UNIVERSIDAD DEL NORTE

RESUMEN

Algoritmo para análisis filogenético: UPGMA

Por **Sandra Milena Acero Barraza**

Director de tesis: Ing. Eduardo Zurek Varela PhD.
Departamento de ingeniería de sistemas

Tesis presentada sobre el diseño y la implementación de una interfaz de software en el lenguaje de programación Python, que permita el análisis filogenético basado en el algoritmo UPGMA (Unweighted Pair Group Method using Arithmetic averages).

En esta investigación se toman como punto de partida las teorías y métodos de la filogenética específicamente el método UPGMA para realizar análisis de secuencias de ADN.
El resultado de esta investigación está enmarcado dentro del área de la bioinformática. Una integración de la informática y la biología que establece una relación entre las teorías y métodos de la filogenética y las técnicas de la ingeniería del software y que se complementan en una implementación para análisis filogenético de secuencias de ADN.

El resultado final consiste en una interfaz que permite cargar un archivo de secuencias de ADN analizarlo y obtener como resultado un árbol jerárquico (árbol filogenético). La aplicación permite guardar dos tipos de archivo: un archivo en formato texto con los pasos intermedios del algoritmo y un archivo de imagen con el árbol filogenético.

TABLA DE CONTENIDO

pág.

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

AGRADECIMIENTOS

El autor desea expresar su más sincero agradecimiento al ingeniero Dr. Eduardo Zurek director del proyecto por su paciencia y colaboración en el desarrollo y preparación de este manuscrito. Además, manifiesta su agradecimiento especial al Dr. Homero Sanjuán y Dr. Guillermo Cervantes, asesores del proyecto, por la colaboración prestada.

INTRODUCCIÓN

Dada la complejidad inherente al virus de inmunodeficiencia adquirida (VIH), y la gran cantidad de esfuerzo empleado por la comunidad científica para encontrar cura al sida e inmunidad al virus que lo provoca, han surgido alrededor del mundo una serie de investigaciones que aunque no han logrado conseguir la meta perseguida, han aportado una serie de conocimiento que puede resultar útil en ese camino.

Una gran parte de las investigaciones está dirigida hacia la búsqueda de una vacuna preventiva que le permita al cuerpo reconocer y defenderse del VIH. El objetivo es desarrollar una vacuna que sea eficaz; sin embargo la habilidad que posee el virus para mutar ha sido el principal obstáculo para los científicos y estudiosos del tema.

Algunos de los tantos estudios realizados arrojaron como resultado el descubrimiento de 3 tipos virales, que a su vez se han clasificados en subgrupos:

- Grupo M, dividido en 9 subgrupos.
- Grupo O, dividido en 3 subgrupos.
- Grupo N

También se hallo una relación geográfica en el subtipo viral, es decir, que en una región determinada del planeta habrá una alta posibilidad de encontrar un subtipo de virus determinado, o un conjunto de subtipos.

En Colombia se desconoce el subtipo predominante, por lo que tiene gran importancia el desarrollo de la investigación que está llevando a cabo el grupo de investigación en biotecnología y el grupo de Virologia y patologías asociadas de la Universidad del Norte,

en busca de este subtipo. La investigación que pretende determinar los subtipos de VIH presente en nuestro país, está implementando una técnica denominada análisis filogenético.
Esta técnica está siendo usada a nivel mundial cada vez más en la determinación de patrones que permitan una clasificación de las muestras obtenidas en la región.

De este planteamiento surge la motivación de esta investigación. El presente proyecto pretende el diseño y la implementación de un algoritmo que permita realizar la clasificación de secuencias de ADN a través de esta técnica.
En esta investigación se toman como punto de partida las teorías y métodos de la filogenética específicamente el método UPGMA para realizar análisis de secuencias de ADN.
La aplicación es realizada en el lenguaje de programación Python versión 2.5.

VISIÓN GENERAL

La presente monografía está compuesta por 5 capítulos. El primer capítulo comprende las especificaciones del proyecto, los objetivos que se desean alcanzar con la realización del mismo, la justificación, el plan de trabajo y la metodología a aplicar.

El segundo capítulo comprende una reseña histórica de los métodos empleados por el hombre a través de la historia, para clasificar los organismos vivos, hasta llegar al análisis filogenético. A continuación se describen los tipos de métodos para análisis filogenético existentes. Y se proporciona una breve descripción, ventajas y desventajas de tres de ellos.

El tercer capítulo trata de manera más específica el método UPGMA, aportando una descripción del mismo, el algoritmo de implementación propuesto en la presente tesis y una descripción de los árboles generados por el método.

En el cuarto capítulo se presentan las estrategias empleadas para el diseño e implementación en el lenguaje Python. También se describen detalladamente las distintas etapas en las que se desarrolló la aplicación y las respectivas consideraciones en el diseño.

En el quinto capítulo se describen los resultados obtenidos en las pruebas realizadas a la aplicación para medir su desempeño en tiempo de ejecución, además de una comparación con otras herramientas similares.

El sexto capítulo contiene las conclusiones finales de la realización de este proyecto.

CAPÍTULO 1: ANTEPROYECTO

1.1 ESPECIFICACIONES DEL PROYECTO

AUTORA DEL PROYECTO

SANDRA MILENA ACERO BARRAZA

Código: 200001338

Teléfono: 3443381 - 301 2910917

Dirección: Calle. 48 # 67B - 89

E-mail: sacero@uninorte.edu.co

DIRECTOR DEL PROYECTO

Eduardo Enrique Zurek Varela, Ph.D.

Ingeniero de sistemas

E-mail: ezurek@uninorte.edu.co

ASESORES

Homero Gabriel Sanjuán Varela, Ph.D.

Médico

E-mail: hsanjuan@uninorte.edu.co

Guillermo Cervantes Acosta, Ph.D.

Químico farmacéutico

E-mail: guicerva@uninorte.edu.co

1.1.1 Entidades interesadas. Las esntidades interesadas en el resultado de esta investigación son:

- Programa de Medicina de la Universidad del Norte
- Programa de Ingeniería de Sistemas de la Universidad del Norte
- Colciencias
- Grupo de Investigación en Biotecnologías
- Grupo de Virología y Patologías Asociadas

1.1.2 Tiempo y costo estimado. El tiempo y costo estimado de este proyecto es:

Tiempo estimado: un (1) año.
Costo estimado: $20.048.000

1.2 DESCRIPCION DEL PROYECTO

1.2.1 Planteamiento del problema de investigación. En estudios realizados por investigaciones a nivel mundial se ha encontrado una heterogeneidad en la secuencia del genoma del virus de inmunodeficiencia humana (VIH), de la misma manera se ha asociado un subtipo o un conjunto de subtipos a regiones determinadas.

En la actualidad los científicos desconocen los subtipos de virus del VIH presentes en Colombia; la carencia de investigaciones al respecto, la necesidad de satisfacer la pregunta científica y las ventajas que conlleva la respectiva clasificación de las muestras obtenidas en nuestra región, y el reconocimiento del subtipo que prevalece, son algunas de las

respuestas que pueden encontrarse al contar con un software que realice análisis filogenético.

1.2.2 Objetivos. Este proyecto de grado está enmarcado en un macro-proyecto cuyo objetivo a largo plazo es crear un sistema de información que permita al grupo de Virología y Patologías Asociadas de la Universidad del Norte clasificar los subtipos de VIH obtenidos en muestras tomadas en la región Caribe colombiana con el fin de construir el análisis filogenético del virus.

- Objetivo general

Este proyecto de grado tiene por objetivo general diseñar e implementar una interfaz de software que facilite el análisis filogenético basado en el algoritmo UPGMA (Unweighted Pair Group Method using Arithmetic averages).

- Objetivos específicos

Recopilar información referente a la filogenética y sus métodos que permita conocer y establecer el marco teórico, para el diseño del algoritmo UPGMA.

Analizar y estudiar el algoritmo UPGMA (Unweighted Pair Group Method using Arithmetic averages) y definir la estructura en la que se basará el sistema de información.

Implementar el algoritmo para análisis filogenético UPGMA (Unweighted Pair Group Method using Arithmetic averages) de secuencias de ADN.

1.3 HIPOTESIS DEL TRABAJO

La implementación del algoritmo para análisis filogenético UPGMA (Unweighted Pair Group Method using Arithmetic averages) como sistema de apoyo a la investigación "análisis filogenético del virus de la inmunodeficiencia humana (VIH) a partir de virus aislados en la población del caribe" permitirá obtener información de las muestras tomadas y recolectar con el uso continuo de esta herramienta información valiosa para lograr el objetivo de su investigación.

La creación de este sistema proporciona una fuente de información que podrá ser empleada no sólo como apoyo en la investigación ya mencionada, sino también en la realización de nuevas investigaciones que conduzcan a conocer más cerca de la biología del virus, conocimiento que será empleado en la realización de pruebas diagnósticas, estudio de nuevas moléculas antivirales, etc. Y encontrar una cura para la enfermedad o inmunidad al virus del VIH.

1.4 JUSTIFICACIÓN

En la actualidad existe una serie de teorías acerca del análisis filogenético como técnica para la clasificación de organismos basados en la secuencia del genoma y algoritmos que sirven para tal fin. Este proyecto pretende emplear esta teoría para crear una aplicación que permita examinar muestras tomadas en la región Caribe colombiana, a través del análisis filogenético del virus, implementando el algoritmo UPGMA (Unweighted Pair Group Method using Arithmetic averages) que sirve para tal fin.

El diseño e implementación del algoritmo UPGMA, brindará a los investigadores del Grupo de virología y patologías asociadas de la Universidad del Norte y a otras personas

que en el futuro se interesen en el área, una herramienta útil para el estudio de los subtipos de VIH.

1.5 PLAN DE TRABAJO

1. Establecimiento del marco teórico necesario para el diseño y la implementación del algoritmo UPGMA para análisis filogenético.
2. Diseño del algoritmo para análisis filogenético UPGMA.
3. Implementación del algoritmo UPGMA, para análisis filogenético de secuencias de ADN.

1.6 METODOLOGÍA

1.6.1 Tipo de estudio. Debido a que el estudio de los tipos de VIH presente en Colombia tiene pocos antecedentes, el presente proyecto pretende brindar a los investigadores una herramienta complementaria a su trabajo.

Por lo anterior se adopta un tipo de estudio exploratorio, el cual permitirá el desarrollo de un proyecto que se convierte en un aporte al campo de las investigaciones referentes al VIH.

1.6.2 Método de investigación. En el desarrollo de este proyecto se empleará el método de investigación síntesis, debido a que el propósito de esta es interrelacionar los conocimientos tanto del área de la biología y Virologia, como del área de la computación, obtenidos en la fase de recolección de información con el fin de integrarlos en un sistema de información.

1.6.3 Técnicas de recolección de la información. La información necesaria para el desarrollo de nuestro proyecto se obtendrá primeramente de la documentación recopilada por los grupos de investigación (grupo de investigación en biotecnologías y el grupo de virología y patologías asociadas) de la universidad del Norte. Otra técnica que emplearemos, es la búsqueda directa de boletines en línea de entidades y/o universidades que estén realizando estudios relacionados al tema.

1.6.4 Pasos metodológicos. Los pasos a seguir en el desarrollo de la investigación son:

1. Recopilación de la información preliminar acerca de la filogenética y del método UPGMA.

El fin de esta fase inicial es fundamentalmente recolectar información para tener una visión general acerca del tema, esta información será recolectada principalmente de charlas con los asesores del proyecto, teniendo en cuenta su conocimiento acerca del mundo de la Virologia. Además de esta, otra gran fuente de información útil en el desarrollo del proyecto son los artículos médicos y científicos disponibles en grandes bases de datos de librerías e institutos médicos.
Además de la recolección de información del área de la biología también en esta etapa e incluso en futuras consultas se hace necesaria la recopilación de información acerca de temas computacionales.

2. Análisis de la información recopilada en la etapa anterior.

En esta etapa se realizará un análisis de la información recolectada en la fase anterior, de manera que sea posible tener una visión más detallada acerca del objetivo al cual se pretende llegar al finalizar este proyecto.

2. Diseño del algoritmo UPGMA.

Llegada a esta fase ya debe existir un conocimiento acerca del algoritmo filogenético UPGMA (Unweighted Pair Group Method using Arithmetic averages), lo que permitirá, en conjunto con los recursos de hardware y software con los que se dispone definir una técnica de programación y una estructura en la cual se basará el desarrollo del software.

4. Implementación del algoritmo UPGMA en el lenguaje de programación Python.

Teniendo definido el diseño del algoritmo, el objetivo en esta fase es implementarlo en un lenguaje de programación, empleando la técnica de programación establecida en la fase anterior.

5. Elaboración de manuales del sistema de información que implementa el UPGMA como método para análisis filogenético.

En esta etapa se debe reunir y completar la documentación correspondiente al software, que se ha estado produciendo desde la etapa de diseño del mismo.

6. Pruebas al sistema de información resultado de esta investigación.

Una vez terminado la aplicación, se realizarán una serie de pruebas para garantizar el buen comportamiento del software. Estas pruebas se realizarán con datos tomados de bases de datos de secuencias del VIH ya existentes.

7. Elaboración de la monografía.

En esta etapa se redactará de manera ordenada el informe final de la investigación realizada y de los resultados obtenidos.

CAPÍTULO 2: ANÁLISIS FILOGENÉTICO

La constante necesidad del hombre de investigar y conocer acerca de sus raíces y su origen lo ha obligado a dirigir su mirada al ambiente que lo rodea. De esta manera surge la necesidad de determinar un orden para agrupar y clasificar los millones de formas de vida que existen en el planeta tierra.

3.1 RESEÑA HISTÓRICA

El primer método de clasificación que se puede identificar es el hecho de que nuestros antepasados hayan asignado una palabra para nombrar a todos los elementos que se encontraban a su alrededor con el simple fin de identificarlos. Tal distinción le asigna a cada objeto un nombre que lo identificaba.

Aristóteles (S. V a.C.) escribió muchos documentos en los cuales se muestra un método que permite clasificar los seres vivos (plantas y animales) de acuerdo a su utilidad y características observables, como por ejemplo, presencia o ausencia de sangre, forma de locomoción, o forma de reproducción. Esta clasificación permaneció hasta el siglo XVIII[1]

Otro método bastante usado en los años siguientes a Galileo fue el llamado método de clasificación artificial; este se basa en la comparación, agrupando aquellos elementos semejantes por sus características físicas observables.

[1] CECCHI, MARÍA CLAUDIA, GUERRERO-BOSAGNA, CARLOS y MPODOZIS, JORGE. El ¿delito? de Aristóteles. *Rev. chil. hist. nat.* [online]. set. 2001, vol.74, no.3 [citado 30 Enero 2007], p.507-514. Disponible en la World Wide Web: <http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0716-078X2001000300001&lng=es&nrm=iso>. ISSN 0716-078X.

Carl Linneo (1707-1778) el llamado padre de la taxonomía crea el sistema de nomenclatura binomial, identificando las especies con un binomio de vocablos latinos de los cuales uno corresponde al género y el otro a la especie. Con su trabajo en el S. XVIII se dio lugar al sistema de clasificación que se usa en la actualidad.

Y finalmente el método que es aceptado por la comunidad científica en la actualidad es la clasificación natural. Consiste en la clasificación basada únicamente en las relaciones genealógicas de los organismos, es decir, se basa en el parentesco evolutivo y el árbol genealógico de cada especie.

A partir de la presunción de que todos los seres vivos poseen un ADN que identifica sus características y la información de su vida y que es trasmitido por herencia directa a sus descendientes. Durante la replicación del ADN de un organismo a su heredero pueden ocurrir que la copia no sea exactamente igual a su original, este cambio en la información genética es lo que se conoce como mutación y es lo que da lugar a la diversificación de las especies.

En la clasificación, toma gran importancia el concepto de evolución, ya permite agrupar los organismos de acuerdo a la relación evolutiva que exista entre ellos, fundamentado en las mutaciones que pudo sufrir un ancestro hasta llegar a las especies en estudio. Lo anterior implica que todos los organismos están genéticamente relacionados y que cambian (evolucionan) con el pasar del tiempo, dejando una historia que es posible reconstruir e identificar.

Producto de lo anterior surge en la biología el término *filogenia* que no es más que una disciplina que proporciona un método que agrupa buscando similitudes entre especies evolutivamente hablando, tomando como referencia la información contenida en la

secuencia de ADN y/o proteínas de los organismos a analizar, representando esto gráficamente a través de los árboles jerárquicos.

Hacia la década de 1930 surge la *escuela sistemática*. Esta escuela se divide en tres corrientes la sistemática evolucionista cuyo representante es Mayr (1930), la sistemática fenética cuyo representante es Sokal (1960) y la sistemática cladística cuyo representante es Henning (1960).

El objetivo principal de la sistemática consiste en reconstruir un esquema gráfico (árbol filogenético) que permita agrupar y clasificar evolutivamente los organismos vivos.
Los pasos a seguir para lograr este objetivo son:

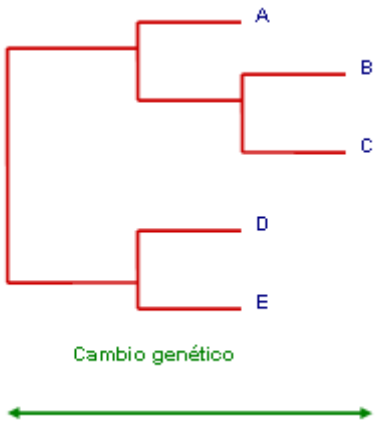
- Escoger las OTUs.
- Identificar las características que se tendrán en cuenta para agrupar, así como, los estados posibles de cada atributo.
- Analizar las características y formular una hipótesis sobre la homología que exista entre ellas.
- Construir una representación gráfica para la clasificación, esto es, representar la hipótesis formulada en el paso anterior en un esquema.

Sistemática evolucionista

Sus principales representantes son Mayr et. al. (1930) y Simpson (1961). Se basa en las relaciones evolutivas y en la semejanza o divergencia de atributos entre dos taxones para clasificarlos. Para esto, tiene en cuenta factores como la zona geográfica donde se desenvuelven los organismos a clasificar y el número de especies por taxón. Para determinar la homología se le asigna un grado de relevancia a cada atributo.

El resultado gráfico de esta clasificación se le denomina *filograma,* Ver figura 1. Y representa tanto la genealogía, como la semejanza y divergencia de los atributos de los taxones.

Figura 1. Filograma



Sistemática fenética

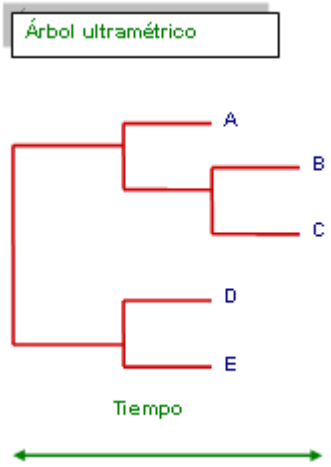
Sus principales representantes son Sokal (1973) y Sneath. La clasificación se basa en la semejanza o divergencia de todos los atributos de las OTU's, cada atributo toma igual importancia en la comparación, por esto entre más información se tenga sobre los taxones, más atributos se contemplan y mejor será la clasificación resultante. La similitud o divergencia en los atributos se calcula a través de cálculos matemáticos, por esto, y por la gran cantidad de información que se debe procesar toma gran importancia el uso del computador como herramienta. Para calcular la semejanza entre dos taxones no existen reglas definidas, sino que se escogen niveles arbitrarios de similitud.

Entre los métodos de análisis filogenético de la corriente fenética se pueden mencionar:

- UPGMA (Unweighted Pair Group Method using Arithmetic averages)
- Neighbor Joining

El resultado gráfico de esta clasificación se le denomina *fenograma,* Ver figura 2. Y representa la semejanza y divergencia global de los taxones.

Figura 2. Fenograma



Sistemática cladística

Su principal representante es Henning (1960). Sus métodos de clasificación se basan en la genealogía, es decir, en las relaciones evolutivas de las OTU's. El análisis cladístico consiste en identificar los atributos que se tendrán en cuenta para clasificar, establecer una hipótesis acerca de los cambios evolutivos de los atributos en las OTU's analizadas, y por último construir un árbol filogenético que represente los cambios de dichos atributos en las OTU's. El concepto principal empleado para clasificar en la cladística, es la apomorfía, que no es más que las nuevas características genéticas que surgen en un grupo de OTU's con un origen común. La cladística utiliza la apomorfía para agrupar las OTU's buscando un ancestro común.

Entre los métodos de análisis filogenético de la corriente cladística se pueden mencionar:

- Parsimony

- Maximum likelihood

El resultado gráfico de esta clasificación se le denomina *cladograma,* Ver figura 3. Y representa la semejanza y divergencia global de los taxones.

Figura 3. Cladograma

3.2 MÉTODOS DE ANÁLISIS FILOGENÉTICO

Resultado de las corrientes de la sistemática han surgido una variedad de métodos para análisis filogenético.

En general existen dos tipos de métodos:

- Métodos basados en caracteres
- Métodos basados en distancia

2.2.1 Métodos basados en caracteres. Los métodos basados en caracteres son aquellos que usan directamente las secuencias (secuencia de ADN o secuencia de proteína) de los taxones para determinar la relación con el ancestro más parecido. En general los métodos cladísticos son basados en caracteres.

- Método Parsimony

Es el método más popular de la sistemática cladística. Reconstruye un árbol filogenético (Cladograma) que representa la relación evolutiva de las especies en estudio. En la construcción del cladograma el método parsimony busca la representación que tenga el menor número de cambios genéticos con respecto a la secuencia de un ancestro común. Evalúa todos los posibles árboles que representen la evolución y busca el más óptimo.

Ventajas

- Es un método de fácil comprensión, ya que sólo analiza los posibles árboles que representen las sustituciones requeridas para un conjunto de secuencias, hallando el que tenga menor cantidad de estas sustituciones.
- De su aplicación resulta un árbol y las hipótesis de la evolución de una característica.

Desventajas

- Requiere de gran trabajo computacional, su complejidad es O(nmk), donde n es la cantidad de taxones, m el numero de características y k la cantidad de valores que pueden tomar las características. "La razón por la que este método requiere gran trabajo computacional es porque el número de posibles árboles que deben ser examinados (para hallar los que posean el mínimo número de sustituciones) es grande, aun para pocas secuencias"[2] .

2.2.2 Métodos basados en distancia. Los métodos basados en distancia primero calculan la distancia total entre todas las parejas de taxones, teniendo en cuenta las

[2]JOHN SOURDIS, AND MASATOSHI NEI. Relative Efficiencies of the Maximum Parsimony and Distance-Matrix Methods in Obtaining the Correct Phylogenetic Tree. Center for Demographic and Population Genetics, The University of Texas Health Science Center at Houston. Junio 25, 1987; 14 páginas.

diferencias en sus secuencias, y luego, calculan un árbol basado en esas distancias. En general los métodos fenéticos son basados en distancia

Todos los organismos poseen características específicas que permiten hallar semejanza o desemejanza con otros organismos, esta comparación se puede lograr basada en la comparación de sus secuencias. La presencia, ausencia o diferencia de un par base en un lugar específico de la secuencia puede determinar su semejanza o diferencia con respecto a la característica que represente dicho par. El término distancia se refiere a una medida resultante de la comparación de las secuencias, con respecto a la frecuencia y orden de los pares en estas. Existen diferentes métodos para hallar esta distancia: distancia de Hamming, distancia de Levenshtein son algunos de ellos. La distancia de Hamming es el número de pares de base que deben cambiarse para transformar una secuencia en otra. Se denomina así por su inventor Richard Hamming (1915-1998). La distancia de Levenshtein es el mínimo número de cambios en una secuencia para ser igual a otra, por borrado, inserción o sustitución.

- Neighbor Joining

Es un caso especial de otro método denominado descomposición de la estrella. Este método utiliza una matriz de distancias e inicialmente el árbol es una estrella. A continuación se reconstruye la matriz de distancias. La separación de cada par de nodos se determina teniendo en cuenta su distancia con respecto al resto de nodos. El árbol se construye teniendo en cuenta la menor distancia entre dos hojas de acuerdo a la nueva matriz de distancias. El árbol generado por este método es un árbol sin raíz.

Ventajas

- Los cálculos realizados sobre los datos (matriz de distancias) son simples, lo que contribuye a que el método sea computacionalmente rápido, su complejidad es $O(n^3)$.
- Es comparativamente rápido con respecto a otros métodos de análisis filogenético.

Desventajas

- Da como resultado un único posible árbol.
- No considera ancestros intermedios entre los nodos existentes.
- Las secuencias no son consideradas como tales, sino que se trabaja con la matriz de distancias lo que puede causar pérdida de información.

- UPGMA

Este método asume que las especies son grupos por si mismas, luego relaciona los dos grupos más cercanos basado en la matriz de distancias, recalcula la matriz de distancia y repite el proceso hasta que todas las especies estén conectadas a un único grupo. El método UPGMA realiza todos sus cálculos con la matriz calculada hallando la distancia genética entre las OTUs.

Ventajas

- Es un método muy sencillo

- Al realizar los cálculos basados en la matriz de distancias y no directamente sobre las secuencias es mucho más rápido computacionalmente que los métodos basados en carácter, tiene una complejidad de $O(n^2)$.

Desventajas

- Las secuencias no son consideradas como tales, sino que se trabaja con la matriz de distancias lo que puede causar pérdida de información.

CAPÍTULO 3: ALGORITMO UPGMA

En el capítulo anterior se presentó una descripción general del método UPGMA, en este capítulo se desarrolla una descripción más específica del mismo. Así, como también se presenta una propuesta del algoritmo, y se explica mediante un ejemplo el funcionamiento del método.

3.1 DESCRIPCIÓN

UPGMA (Unweighted Pair Group Method using arithmetic Averages)

Método para análisis filogenético definido por Peter H. A. Sneath y Robert R. Sokal 1973., principales representantes de la escuela fenética. Es un algoritmo heurístico que usualmente encuentra una solución muy acertada.

Consiste en la búsqueda de la distancia más pequeña en la matriz de distancias genéticas y agrupar las unidades que la conforman como una sola unidad taxonómica independiente.

Se calculan los promedios de la nueva unidad contra las restantes creando una nueva matriz y se repite el proceso hasta que todas las unidades queden unidas a un único elemento (ancestro hipotético).

El método UPGMA al igual que todos los demás métodos fenéticos basados en distancia, dadas las secuencias genéticas de los taxones, genera una matriz de distancia y realiza sus cálculos con esta. La matriz de distancia *matriz* es una matriz cuadrada de tamaño *n*, donde *n* es el número de taxones a clasificar y *matriz*[*i*][*j*] es la distancia genética entre una especie *i* y una especie *j*.

La matriz de distancia calculada debe poseer 3 características:

- La matriz de distancia *matriz* debe ser métrica. Una matriz es métrica si satisface:
  - Simetría: $matriz[i][j] = matriz[j][i]$ y $matriz[i][i] = 0$
  - Desigualdad triangular: $matriz[i][j] + matriz[j][k] \geq matriz[i][k]$
- La matriz de distancia *matriz* debe ser métrica aditiva. Una matriz es métrica aditiva si se cumple que existe un árbol donde:
  - Cada rama tiene un peso positivo y cada hoja corresponde a una especie.
  - $\forall i,\ j,\ 1 \leq i \leq n,\ i < j \leq n,\ matriz[i][j]$ es la suma de los pesos de las ramas desde la hoja *i* hasta la hoja *j*. Dicho árbol también es llamado árbol aditivo.
- La matriz de distancia *matriz* debe ser ultramétrica. Una matriz es ultramétrica si:
  - Es métrica aditiva
  - La raíz del árbol que se forma a partir de la matriz de distancia *matriz*, y de todos los subárboles que contiene, es tal que la suma de todos los pesos de las ramas salientes de esta es la misma. Dicho árbol, también es llamado ultramétrico.[3]

[3] NING K., SHAN T., XIANG S. L., SHEN W. Phylogenetic Tree Reconstruction: Distance Based. . [online]. Octubre 10 de 2003. Disponible en: <http://www.comp.nus.edu.sg/~bioinfo/phylogenetic%20tree%20reconstruction_2_8.pdf> . Fecha de consulta: 13 de febrero de 2007. 20 páginas.

3.2 ALGORITMO

Entrada: n secuencias de ADN o proteína las cuales serán llamadas OTUs o taxones.

Salida: El fenograma que representa la relación evolutiva existente entre las n OTUs.

Algoritmo:

Sea $OTUS = \{t1, t2, t3, ..., tn\}$ el conjunto de las n secuencias, donde cada ti representa una especie.

Sea $MatDistancia$ una matriz cuadrada de tamaño n.
Repetir para $i = 1, ..., n$
  Repetir para $j = i, ..., n$
    Calcular $dist(ti, tj)$
    $MatDistancia[i][j] = dist(ti, tj)$
    $MatDistancia[j][i] = dist(ti, tj)$
    $MatDistancia[i][i] = 0$
  Fin repetir
Fin repetir
Repetir n-1 veces
  Buscar ti, tj tal que $dist(ti, tj)$ en $MatDistancia$ sea mínima.
  Hacer $f = i, c = j$.
  Definir un nuevo $tk = ti \cup tj$ .En $OTUS$
  Reemplazar ti y tj por tk
  Recalcular la matriz de distancia.
  Hacer $n = n - 1$, tamaño de matDistancia igual a n
  Repetir para $i =, ..., n$
    Repetir para $j = i, ...n$
      Si $i \neq f \wedge i \neq c \wedge j \neq f \wedge j \neq c$:
        $matDistancia[i][j] = dist(ti, tj)$
      Fin Si

```
        Si  j = f :
            matDistancia[i][j] = (dist(i,tf) + dist(i,tc)) / 2
        Fin si
        Si  i = f :
            matDistancia[i][j] = (dist(c,tj) + dist(f,tj)) / 2
        Fin si
        matDistancia[i][j] = matDistancia[j][i]
        matDistancia[i][i] = 0
      Fin repetir
   Fin repetir
   Hacer  distancia(k) = dist(f,tc) / 2
   Crear Subárbol uniendo rama tf con rama tc , con una distancia de dist(k)[1]
Fin repetir
```

3.3 GENERACIÓN DEL ARBOL

El árbol generado al aplicar el método UPGMA sobre un grupo de n taxones, debe ser ultramétrico y por lo tanto aditivo. Debido al carácter fenético del método a este se le denomina fenograma.

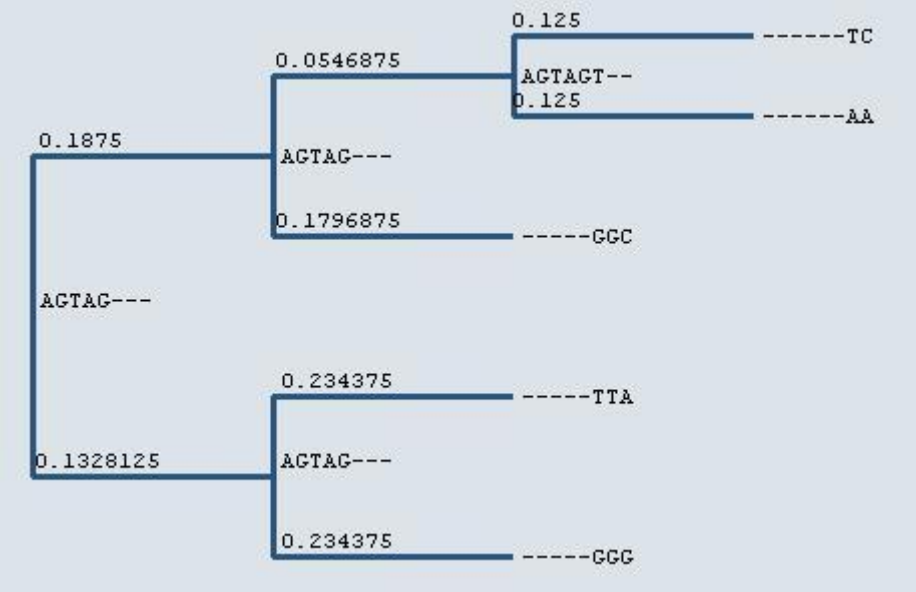
Este árbol representa la relación evolutiva de n taxones agrupando de acuerdo a la similitud de sus secuencias por medios de ancestros comunes hasta llegar a un ancestro común a todos. Ver figura 4.

Cada hoja del árbol representa un taxón, por tanto existen tantas hojas como taxones se estén estudiando. Éstas están etiquetadas con la(s) partes de la secuencia que caracteriza y diferencia cada OTU de las demás.

La unión de dos hojas forma una nueva unidad taxonómica independiente o ancestro común hipotético entre las especies en cuestión. Una Hoja puede estar unida a otra o a un subárbol que representa un ancestro común a otros taxones previamente asociados.

Cada unión de hojas con hojas, hojas con subárboles o subárboles con subárboles tiene una etiqueta que representa la(s) partes de la secuencia que tienen en común las especies que se están asociando. Además de esto, cada rama posee un peso que representa la distancia genética entre las especies y que resulta de los cálculos realizados en la matriz de distancias.

Figura 4. Características del fenograma



3.4 EJEMPLO

En el siguiente ejemplo se muestra el cálculo de la matriz de distancias, de acuerdo a la implementación propuesta y la generación del árbol a partir de esta:

PASO 1: Crear la matriz de distancia

$$OTUs = \begin{bmatrix} AGTAGTTC \\ AGTAGTTA \\ AGTAGTAA \\ AGTAGGGG \\ AGTAGGGC \end{bmatrix}$$

1. AGTAGTT**C**
2. AGTAGTT**A**

$$dist(1,2) = \frac{1}{8} = 0.125$$

1. AGTAGTT**C**
3. AGTAGTT**A**

$$dist(1,3) = \frac{2}{8} = 0.25$$

Tabla 1. Matriz de distancia por distancia de Hamming

OTU	1	2	3	4	5
1	0	0.125	0.25	0.375	0.25
2	0.125	0	0.125	0.375	0.375
3	0.25	0.125	0	0.375	0.375
4	0.375	0.375	0.375	0	0.125
5	0.25	0.375	0.375	0.125	0

Aplicando la fórmula diseñada por Tajima obtenemos la siguiente matriz:

Tabla 2. Matriz de distancia por factor de corrección Tajima

OTU	1	2	3	4	5
1	0	0.875	1.375	1.7453	1.375
2	**0.875**	0	0.875	1.7453	1.7453
3	1.375	0.875	0	1.7453	1.7453
4	1.7453	1.7453	1.7453	0	0.875
5	1.375	1.7453	1.7453	0.875	0

PASO 2: Hallar la menor distancia en la matriz.

Menor distancia: $dist(1,2) = 0.875$

PASO 3: Unir las Otus con menor distancia.

Unir las Otus 1 y 2. Y calcular su distancia:

$$distancia = \frac{dist(1,2)}{2} = \frac{0.875}{2} = 0.4375$$

Figura 5. Subárbol resultante de unir OTUS 1 y 2

PASO 4: Recalcular la matriz de distancia:

Tabla 3. Matriz de distancia recalculada – Paso 4

OTU	12	3	4	5
12	0	1.125	1.7452	1.5601
3	1.125	0	1.7453	1.7453
4	1.745373	1.7453	0	0.0875
5	1.5601	1.7453	**0.875**	0

PASO 5: Volver al paso 2. Hasta todas las Otus estén relacionadas a través de un ancestro común.

Menor: 0.875
Unir Otus 4 y 5

Figura 6. Subárbol resultante de unir OTUS 4 y 5

Tabla 4. Matriz de distancia recalculada – Paso 5.1

OTU	12	3	45
12	0	1.125	1.6527
3	**1.125**	0	1.7453
5	1.6527	1.7453	0

Menor: 1.125
Otus: 12 y 3

Figura 7. Subárbol resultante de unir OTUS 12 y 3

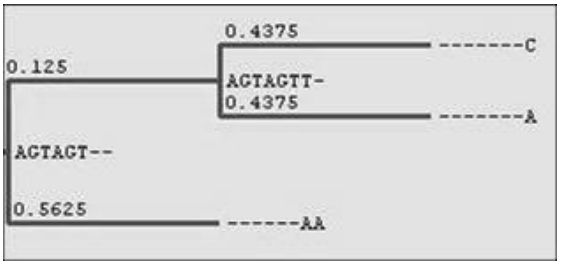
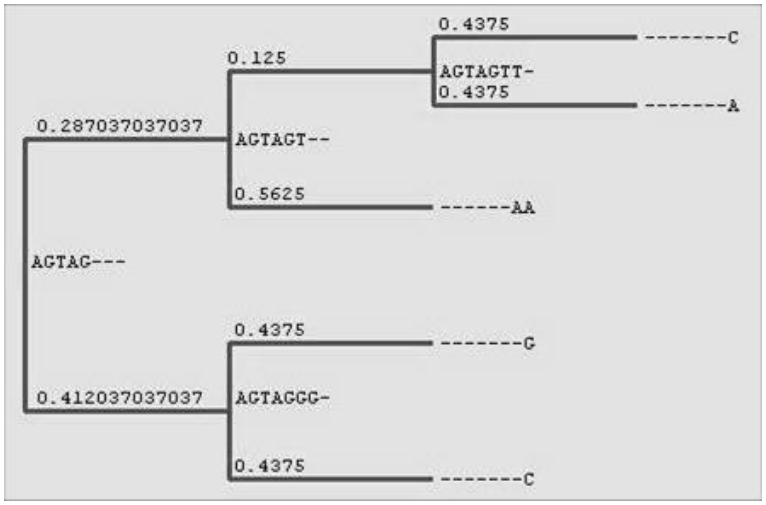


Tabla 5. Matriz de distancia recalculada – Paso 5.2

Otus	123	45
123	0	1.6527
45	**1.6527**	0

Figura 8. Subárbol final



CAPÍTULO 4: DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

En el capítulo anterior se realizó una descripción del algoritmo UPGMA y del árbol generado por este. Además se explicó su funcionamiento a través de un ejemplo. En este capítulo se presentarán las estrategias empleadas para el diseño e implementación en el lenguaje Python.

4.1 DESCRIPCIÓN DE LA IMPLEMENTACIÓN

El proceso realizado por software de implementación del algoritmo para análisis filogenético consta de tres partes:

- Primera parte: Captura de los datos de entrada (Otus).
- Segunda parte: Creación de la matriz de distancia a partir de los datos de entrada, realización de cálculos, agrupaciones, recálculo de la matriz y formación del fenograma.
- Presentación de la salida.

4.2 ETAPAS DE DESARROLLO

El desarrollo del software de implementación del algoritmo para análisis filogenético se lleva a acabo en 4 etapas:

- Primera etapa: Creación de la matriz de distancias a partir de las OTUS. Cálculo de la divergencia genética (distancia) entre cada par de Otus.

- Segunda etapa: Desarrollo del método UPGMA

- Tercera etapa: Captura de los datos de entrada (Otus).

- Cuarta etapa: Diseño y desarrollo de la interfaz gráfica

4.2.1 Creación de la matriz de distancias. Cada OTU (Unidad Taxonómica Operativa) representa una secuencia genética y cada secuencia genética representa una serie de datos de un organismo o especie.

Una secuencia de ADN está dada por una sucesión de caracteres, cada uno de los cuales por separado o en conjunto representan información genética sobre el organismo.

La secuencia de ADN está formada por la combinación de caracteres en la sucesión, los cuales son denominados nucleótidos. Para el ADN existen cuatro tipos de nucleótidos representados por las letras A (Adeline), C (Cytosine), G(Guanine) y T (Thymine).

A continuación dos ejemplos de secuencia de ADN:

Ejemplo 1.
TGCTCTCACATCTTCTTGGCCAGCACTGGACCACACAACTCCTTCTAGATAC
AGAGGAGTCCTAGGATTC

Ejemplo2.
TATGAGAAAGAAGGGGAGGGTGGGCAAAGGGCAGCCAGCTGTGCAGCA
TCTGCTGGAGACACCTAAC

Las secuencias de ADN tienen alrededor de 9000 nucleótidos.

La matriz de distancias es el cálculo de la distancia genética entre cada una de las OTUS a analizar.

El método más común para calcular la distancia en este tipo de implementaciones es empleando el método de Hamming. Este consiste en contar el número de cambios que deben realizarse sobre una secuencia para que sea igual a la otra y dividirlos entre el total de nucleótidos.

Ejemplo: Sean las secuencias:

Secuencia 1. TCTGCTGG**CCTAGGA**T**T**CG

Secuencia 2. TCTGCTGG**AGACACC**T**A**CG

En la secuencia 1 deben realizarse cambios en 8 nucleótidos para que ésta sea igual a la secuencia 2. Por tanto la distancia de Hamming entre estas dos secuencias es: $\frac{8}{19} = 0{,}421052631578947368421052631578 95$

De esta manera, la distancia entre secuencias deducida por Hamming resulta coherente teniendo en cuenta que representa la cercanía o separación entre las especies evolutivamente hablando, y esto está dado por la cantidad de cambios que se han producido en sus informaciones genéticas desde un ancestro común.

Sin embargo, este método falla cuando se deben tener en cuenta las mutaciones múltiples en el mismo sitio (nucleótido o conjunto de nucleótidos). A esto se le denomina homoplasia, que en otras palabras son características de diferentes especies que se parecen pero no tienen el mismo origen, es decir que la adquisición de un mismo carácter en dos organismos no es causa de que estos tengan una descendencia común.

La solución a este problema fue inicialmente resuelta por Jukes y Cantor[4], los cuales propusieron un factor de corrección, dado por la siguiente fórmula:

$$k(A,B) = -\frac{3}{4}\log_e\left[1-\frac{4}{3}\overline{dist}(A,B)\right]$$

Donde $k(A,B)$ es la distancia corregida y $\overline{dist}(A,B)$ es la distancia de Hamming.

Esta fórmula es válida para $\overline{dist}(A,B) < 0.75$, ya que de otra manera es imposible el cálculo del logaritmo.

Para dar solución a este nuevo inconveniente que surgió, Tajima y Nei diseñaron una nueva fórmula [5]:

$$D(A,B) = \sum_{i=1}^{k} \frac{k^{(i)}}{i\left(\frac{3}{4}\right)^{i-1} m^{(i)}}$$

Donde m es el total de pares de base que poseen las secuencias, k es el número de pares base que difieren en las dos secuencias, luego $k = m - dist(a,b)$

$$k^{(i)} = \frac{k!}{(k-i)!}$$

$$n^{(i)} = \frac{n!}{(n-i)!}$$

A continuación se describe parte de la implementación en Python del algoritmo aplicando la fórmula de Tajima.

```
def calcularMatriz (self, n,m, OTUS):
        """Utilizando el modulo matrices crea una matriz de tam. nxn
        Calcula la matriz de distancias basado en el método de distancia de Hamming.

        dist(A,B)= Total de cambios/longitud de las OTU's
        D(A,B)= sumatoria de i=1,k de (k**(i))/(i*((3/4)**(i-1))*(m**(i)))

        Recibe como argumentos el tamaño de la matriz la longitud de las secuencias
         y las secuencias
```

[4] JUKES, T.H., y C.R. CANTOR. 1969. Evolution of Proteins Molecules. Pp 21-132 en H:N: MUNRO, ed. Mammalian protein metabolism. Academis Press, New York.

[5] TAJIMA, Fumio. MSATOCHI, Nei. Unbiased Estimation of Evolutionary Distance between Nucleotide Sequences. Department of population Genetics, Nacional Institute of genetics. University of Chicago. Mol. Biol. Evol. 10(3). Pp. 677-688. Octubre 6 de 1993.

```
y devuelve la matriz de distancias en listas anidadas (matDistancia)"""
self.matDistancia = matrices.create((n),(n))
for i in range(0,n-1,1):
    for j in range(i+1,n,1):
        self.dist = 0.0
        self.otu1 = OTUS[i]
        self.otu2 = OTUS[j]
        for k in range(0,m,1):
            if self.otu1[k] != self.otu2[k]:
                self.dist = self.dist + 1


        # APLICANDO METODO DESARROLLADO POR TAJIMA
        # Tajima (1993, MBE 10:677-688)

        self.s = 0.0
        self.d = (int)(self.dist/m)                                          (1)
        for l in range(1,self.d+1,1):
            self.num = (float)(self.factorial(k))/(self.factorial(k-l))      (2)
            self.den = (float)(l*((3.0/4.0)**(l-1.0)))*((self.factorial(m))
                                /(self.factorial(m-l)))                      (3)
            self.s += (self.num/self.den)
        if self.d == 1.0:
            self.num = (float)(self.factorial(k))/(self.factorial(k-1))
            self.den = (float)(1*((3.0/4.0)**(0)))*((self.factorial(m))
                                /(self.factorial(m-1)))
            self.s = (self.num/self.den)

        self.dist = self.s


        self.matDistancia[j][i] = self.dist
        self.matDistancia[i][j] = self.dist
return self.matDistancia
```

En (1) d se inicializa con el valor de distancia calculado originalmente con el método de Hamming. En (2) se calcula el numerador. En (3) se calcula el denominador de la fórmula.

La matriz de distancias es una matriz cuadrada de nxn, donde n es la cantidad de secuencias a analizar.

$dist(A,B) = dist(B,A)$ y $dist(A,A) = 0$, por tanto la matriz de distancias es simétrica, y la diagonal principal es cero.

4.2.2 Desarrollo del método UPGMA. Una vez formada la matriz de distancias, el paso a seguir es ejecutar sobre esta matriz el método descrito en el capítulo anterior.

La estructura de datos empleada para la representación del árbol resultado del proceso de análisis es una lista. En el lenguaje Python una lista es una secuencia mutable. Una secuencia es un conjunto de elementos, por ejemplo: una cadena de caracteres es una secuencia y sus elementos son los caracteres. Las secuencias pueden ser mutables e inmutables. Una secuencia mutable es aquella en la cual sus elementos pueden ser cambiados. En una secuencia inmutable los elementos no se pueden cambiar. En python las cadenas son secuencias inmutables, en otras palabras, no se pueden cambiar. Por ejemplo:

```
>>> palabra="python"
>>> palabra[0]
'P'
>>> palabra[0]="s"

Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    palabra[0]="s"
TypeError: 'str' object does not support item assignment
>>>
```

En el anterior ejemplo se puede observar como teniendo la variable palabra que representa una cadena de caracteres, es imposible cambiar el elemento ubicado en la posición cero. Esto demuestra la característica de inmutabilidad de las cadenas.

Las listas por su parte son mutables, lo que las hace más flexibles para la modificación y eliminación de elementos.

Una lista tiene la siguiente forma: $nombre = [elemento1, elemento2, ..., elementok]$ y al igual que cualquier secuencia se puede acceder a sus elementos por indexación. La lista es una estructura poderosa en el sentido que puede contener elementos de distintos tipos en una misma lista e inclusive puede contener listas enteras, a este tipo se le denomina listas anidadas.

Para representar el árbol filogenético se empleó una lista anidada (self.arbol) que además de otras listas también contiene elementos numéricos y cadenas de caracteres.

El formato es el siguiente:

$$self.arbol = [subarbol1, subarbol2, distancia, cadenaComun, distaRama1, distRama2]$$

Donde *subarbol1* y *subarbol2* pueden ser:

$$(1) subarbol = [subarbol1, subarbol2, distancia, cadenaComun, distaRama1, distRama2]$$
$$(2) subarbol = [numerodeSecuencia, subarbol2, distancia, cadenaComun, distaRama1, distRama2]$$
$$(3) subarbol = [subarbol1, numerodeSecuencia, distancia, cadenaComun, distaRama1, distRama2]$$
$$(4) subarbol = [numerodeSecuencia, numerodeSecuencia, distancia, cadenaComun, distaRama1, distRama2]$$
$$(4) subarbol = numerodeSecuencia$$

(1) Un subárbol cuyas dos ramas son también subárboles.
(2) Un subárbol donde primera rama es una hoja y la segunda rama es un subárbol.
(3) Un subárbol donde la primera rama es un subárbol y la segunda rama es una hoja.
(4) Un subárbol donde la primera y la segunda rama son hojas.
(5) Una hoja (*numerodeSecuencia*)

Además, *distancia* representa la distancia de la matriz de distancias que representa la unión de los subárboles representados.

CadenaComun representa los elementos de las secuencias similares en la agrupación.

distRama1 representa el valor de distancia asociado a la rama uno. Y *distRama2* representa el valor de distancia asociado a la rama dos.

Ejemplo de representación de un árbol:

OTU 0 AGTAGTTC

OTU 1 AGTAGTTA

OTU 2 AGTAGTAA

$$arbol = [[0, 1, 0.4375, 'AGTAGTT-', 0.4375, 0.4375], 2, 0.5625, 'AGTAGT--', 0.125, 0.5625]$$

Subárbol1 es $[0, 1, 0.4375, 'AGTAGTT-', 0.4375, 0.4375]$ que representa un subárbol del tipo (4).

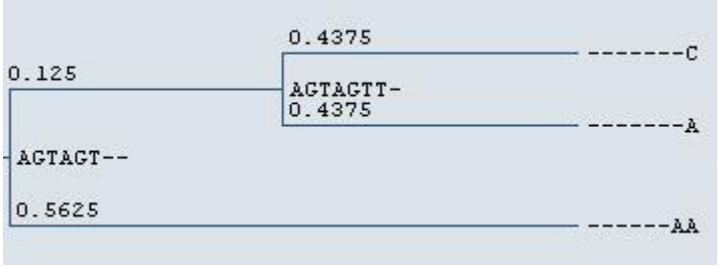
Subárbol2 es 2 que representa la OTU 2

distancia entre *subárbol1* y *subarbol2* es 0.5625

La *cadenaComun* es 'AGTAGT—'

La *distRama1* es 0.125 y *distRama2* es 0.5625

Figura 9. Ejemplo de representación del árbol filogenético.



A continuación se presenta el código de la función que ejecuta el método UPGMA y forma el árbol con el formato ya descrito.

En (1) se inicializa el árbol con los números de 0 a n, donde n es el número de secuencias que contiene el archivo. Estos números representaran las secuencias dentro del árbol..

El ciclo (2) representa una iteración mientras que existan secuencias que no hayan sido agrupadas.

En el ciclo se realizan tres pasos: hallar la menor distancia en la matriz, recalcular la matriz y formar el árbol.

Y por último al salir del ciclo, el paso a seguir es dibujar el árbol ya formado.

```
def ejecutar(self):
        #Ejecuta el método UPGMA
         self.arbol = []
        for i in range(0,self.n,1):         (1)
            self.arbol.append(i)

        #Esta lista representa los caracteres diferentes de cada OTU
```

```
            self.sim = self.OTUS[:]

            #Iterar mientras que las OTU's no se hayan unido
            self.nTemp = self.n
            while self.nTemp > 1:               (2)

                #   1.   Hallar la menor distancia en la matriz de distancias
                self.menor, self.imenor, self.jmenor =
self.control.hallarMenorDistancia(self.matDistancia, self.nTemp)


                #   2.   Recalcular matriz
                self.matDistancia, self.nTemp =
self.control.recalcularMatriz(self.matDistancia, self.nTemp,  self.imenor, self.jmenor)


                #   3.   Formar arbol
                self.arbol, self.sim = self.control.formarArbol(self.arbol, self.imenor,
self.jmenor, self.menor, self.m, self.sim)


                if len(self.arbol)==1:
                    self.arbol=self.arbol[0]

            #   5.  Dibujar arbol
            self.dibujarAr.dibujar(self.arbol, self.n, self.OTUS, self.m)
```

4.2.3 Captura de datos de entrada. En el proceso de diseño se determinó que el método más apropiado para la captura de los datos de entrada era a través de un archivo.

El archivo debe contener las secuencias a analizar.

En el mundo de la bioinformática existen muchos tipos de formatos de representación de secuencias. Uno de los más usados y sencillos es el formato Fasta.

Fasta es un formato basado en texto usado para representar secuencias de ADN, en las cuales cada par base es representada por un código. El formato también permite escribir nombres de las secuencias y una corta descripción.

El archivo empleado por la aplicación está basado en el formato Fasta, y tiene extensión: .txt. El archivo está formado como sigue:

>CódigoIdentificador Descripción corta|len=(entero)longitud[Enter]

Secuencia (A,C,G,T) [Enter]

[Enter]

>>CódigoIdentificador Descripción corta [Enter]

Secuencia (A,C,G,T) [Enter]

[Enter]

…

La representación de una secuencia consta de dos partes:

- Encabezado
- Información de las secuencias

Encabezado:

Debe tener una longitud menor o igual a 80 caracteres.
Inicia con el símbolo > seguido del identificador de la secuencia sin espacio.
El encabezado de la primera secuencia debe tener la longitud de estas precedida del símbolo | y la palabra len seguida de =.

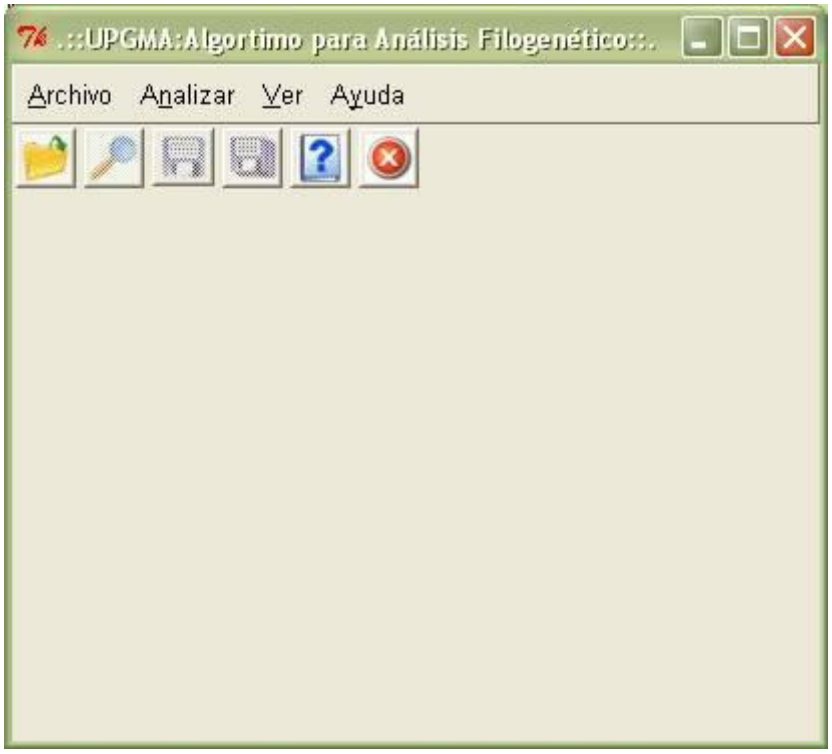
Información de las secuencias:

Son las secuencias en si. Están representadas por secuencias de letras (A = Adeline, C = Cytosine, G = Guanine y T = Thymine)
Cada secuencia debe estar separada de la siguiente por una línea de espacio <Enter>.
Al final de la secuencia final debe haber una línea de espacio <Enter>

El programa posee un módulo destinado a las operaciones realizadas sobre el archivo, como son lectura y validación del formato y de las secuencias contenidas.

4.2.4 Diseño y desarrollo de la interfaz gráfica. La interfaz de usuario de la aplicación se presenta en un entorno similar al de Windows por lo cual cualquier usuario familiarizado con Windows se sentirá también muy familiarizado con el ambiente de la aplicación al manejarla.

Figura 10. Interfaz gráfica de la aplicación.



Para el diseño de esta interfaz se empleó Tkinter, que no es más que un conjunto de módulos que sirven de herramienta estándar de Python para el desarrollo de la Interfaz Gráfica de Usuario de los programas. Tkinter fue formalmente desarrollada por Sun Labs.

Al igual que Python es una herramienta de código abierto y se instala por defecto en la instalación de Python.

Además de esto se empleó el módulo Pmw, que es una herramienta para el diseño de componentes gráficos de alto nivel. Pmw trabaja usando en módulo Tkinter[6].

Pmw consiste en un conjunto de clases base y una librería flexible y extensible de componentes denominados, megawidgets. Entre estos megawidgets se incluyen comboboxes, cuadros de diálogo, listas, radioButtons, Elementos con Scroll, etc.

La ventana de la aplicación consta de 5 áreas:

1. Barra de menú: Se encuentra en la parte superior, guarda el formato de las barras de menú estándar de Windows.

2. Barra de herramientas: Se encuentra debajo de la barra de menú, contiene atajos a las principales opciones de la barra de menú. Las imágenes empleadas en el diseño de los íconos hacen parte de un conjunto de iconos para barras de herramientas disponibles de manera libre para desarrolladores de software en: http://www.neatui.com/. Para no hacer más pesado el programa con la carga de las imágenes, estas se encuentran embebidas en el código. Para realizar esto se empleó el script Image Embedder 1.0 que genera el código base64 de la imagen[7].

[6] Paquete: Pmw . Release: 1.2 . Date: agosto 4 2003. URL: http://sourceforge.net/projects/pmw/. Fecha de descarga: Marzo 7 de 2007 9:46 a.m. Administrador del proyecto: Greg McFarlane. Sistema Operativo: OS Independen (Escrito en un lenguaje interpretado). Licencia: MIT License. Category: Graphics, Software Development

[7] Script: Image Embedder 1.0 . Release: 1.0 . Fecha: 26 Feb 01. URL: http://www.3dartist.com/WP/python/pycode.htm#img2pytk. Fecha de descarga: 01:27 p.m. 24/05/2007.. Autor: Hill Allen. Email del autor: dempy@3dartist.com Operating System: OS Independent (Written in an interpreted language). Licencia: GNU License. Categoria: Graphics, Software Development

3. Área de visualización de las secuencias: Esta área se activa luego de abrir un archivo. Contiene un TextArea con las secuencias que contiene el archivo. Esta área dispone de una barra de desplazamiento que permite visualizar la información en su totalidad.

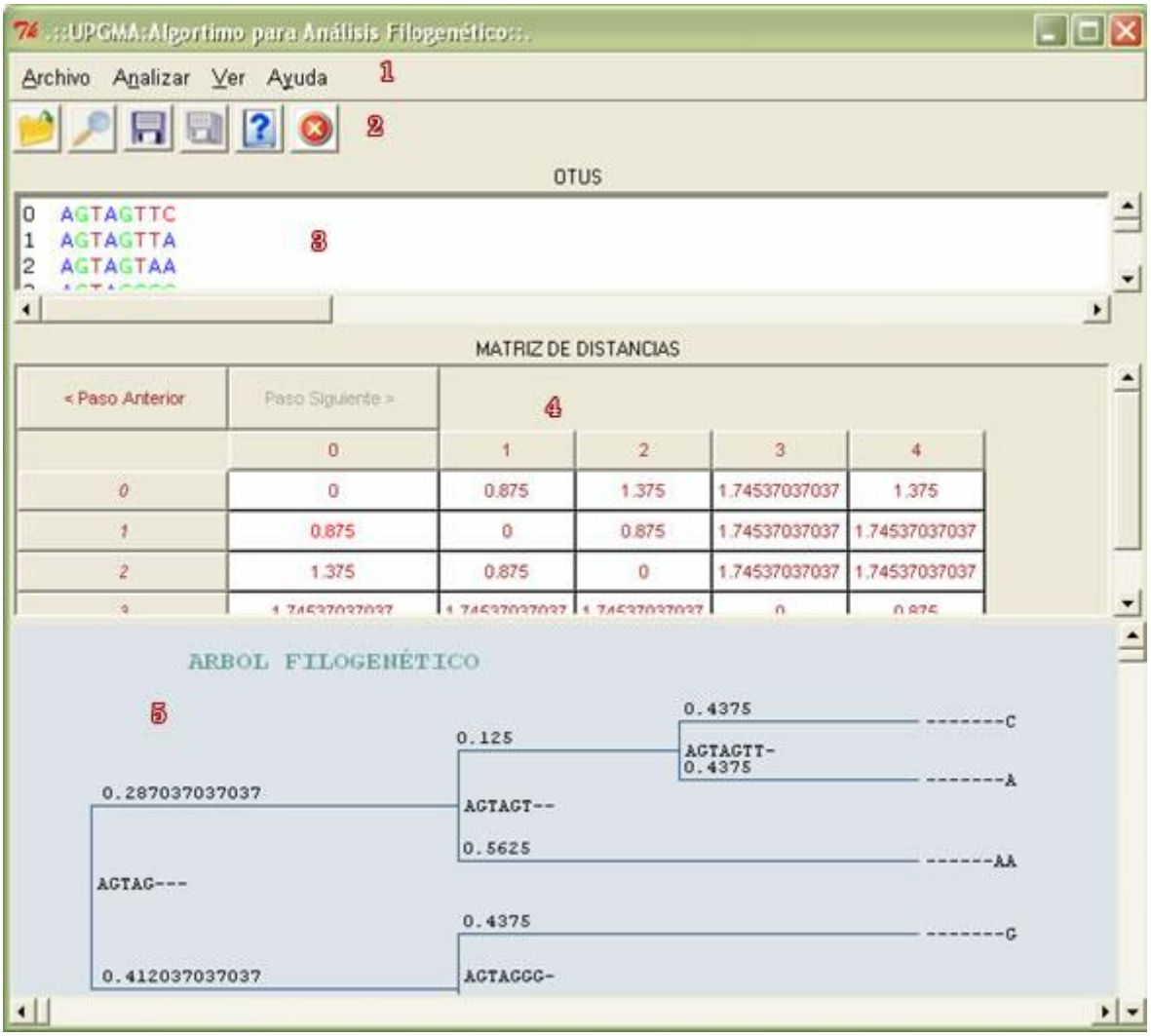
4. Área de visualización de la matriz de distancia: Esta área se activa luego de abrir un archivo. Contiene dos botones (paso anterior y paso siguiente) que permiten ver los cambios en la matriz de distancia en cada iteración del algoritmo, sólo se activan luego de analizar las secuencias. Esta área también contiene la matriz. Esta área dispone de una barra de desplazamiento que permite visualizar la información en su totalidad.

5. Área de visualización del árbol.: Esta área se encuentra en la parte inferior de la ventana y se puede visualizar únicamente después de analizar las secuencias. Muestra el árbol filogenético resultado del análisis. Esta área dispone de una barra de desplazamiento y características que permiten hacer zoom in y zoom out.

La interfaz gráfica permite guardar una imagen en formato jpg del árbol resultante. Para el manejo de imágenes en Python se empleó la librería Python Imaging Library (PIL). Esta librería le brinda a Python la capacidad de procesar imágenes. Soporta formatos de imagen como gif y jpg[8].

[8] Versión: PIL 1.1.6. disponible en: http://www.pythonware.com/products/pil/index.htm. descargado 02:20 p.m. 03/junio/2007. Author: Secret Labs AB (PythonWare). Author_email: info@pythonware.com. Description: Python Imaging Library. Name: PIL. Url: http://www.pythonware.com/products/pil. Versión actual: 1.1.6

Figura 11. Áreas de la ventana de aplicación.



CAPÍTULO 5: PRUEBAS Y RESULTADOS OBTENIDOS

En el capítulo anterior se realizó una descripción de las herramientas y procedimientos empleados en la implementación en Python del algoritmo UPGMA. En este capítulo se presentarán los resultados obtenidos en las pruebas realizadas a la aplicación, además de una comparación con otras herramientas similares.

5.1 CASO GENERAL

Dadas las siguientes 5 secuencias de longitud 20:

Tabla 6. Secuencias del caso general

Número	Secuencia
0	ATGGCTATTCTTATAGTACG
1	ATCGCTAGTCTTATATTACA
2	TTCACTAGACCTGTGGTCCA
3	TTGACCAGACCTGTGGTCCG
4	TTGACCAGTTCTCTAGTTCG

La matriz que representa las distancias genéticas existentes entre ellas es la siguiente:

Tabla 7. Matriz de distancia del caso general

OTU	0	1	2	3	4
0	0	2.52777777778	5.8541871037	5.187503916	4.57708210856
1	2.5277777778	0	4.57708210856	6.57987116357	5.85341871037
2	5.8541871037	4.57708210856	0	2.0537037037	4.57708210856
3	5.187503916	6.57987116357	**2.0537037037**	0	3.00185185185
4	4.57708210856	5.85341871037	4.57708210856	3.00185185185	0

5.2 RESULTADOS

La implementación realizada en el presente trabajo del algoritmo UPGMA ha demostrado que funciona bien al compararse con otras implementaciones como son una implementación de construcción de árboles filogenéticos de Durban et al: Biological Sequence Análisis realizada por Peter Sestoft que es un applet de java que funciona en una página Web y puede ser utilizado en línea[9] y el programa Phylip que es un paquete de programas libres que además del UPGMA contiene otras implementaciones para análisis filogenético. Puede ser utilizado a través de ejecutables instalados en el computador o en línea a través de un explorador de Internet.[10].

5.2.1 Formatos de archivos de entrada. A continuación se describen los formatos de archivo de entrada de las tres aplicaciones.

- UGMA v1.0:

  Como se describe en la sección 4.2.3 del capítulo 4 (Captura de los datos de entrada). La aplicación UPGMA v1.0 recibe como entrada un archivo con formato texto (.txt) basado en el formato Fasta, que contiene las secuencias.

  El archivo de entrada está formado como sigue:

[9] Implementación en Java de loa algoritmos para análisis filogenéticos descritos en el libro Durbin et al: Biological Sequence Analysis, Cambridge University Press 1998, chapter 7. Autor: Peter Sestoft, email del autor: sestoft@itu.dk 1999-12-07, versión: 0.3. Disponible en: http://www.itu.dk/people/sestoft/bsa/Match7Applet.html. Fecha de consulta: mayo 20 2007

[10] PHYLIP. Autores: Akiko , Dan Fineman , Patrick ,Daniel Yek, etc. Distribuido desde 1980. Licencia: código abierto. Página Web: http://evolution.genetics.washington.edu/phylip.html.

>001 secuencia de prueba Número 0|len=20
ATGGCTATTCTTATAGTACG

>002 secuencia de prueba Número 1
ATCGCTAGTCTTATATTACA

>003 secuencia de prueba Número 2
TTCACTAGACCTGTGGTCCA

>004 secuencia de prueba Número 3
TTGACCAGACCTGTGGTCCG

>005 secuencia de prueba Número 4
TTGACCAGTTCTCTAGTTCG

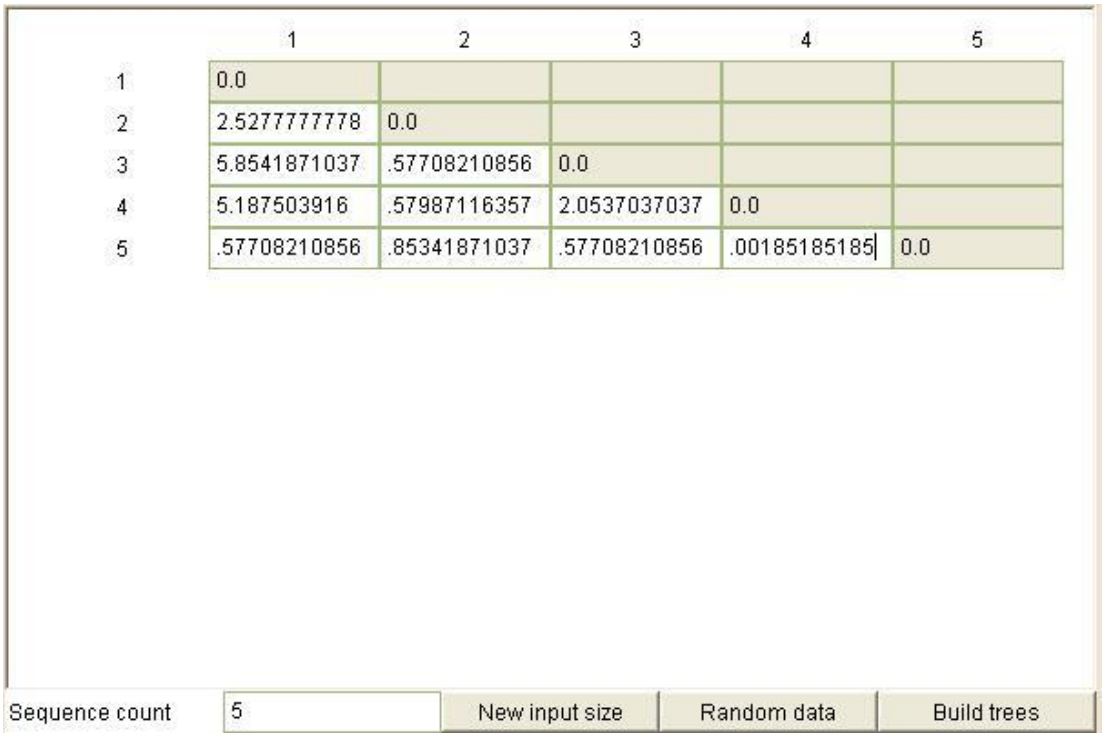
- Implementación de Meter Sestoft:

En la figura 12 se muestra la interfaz que posee por esta implementación. Donde sus principales datos de entrada son: sequence count que representa el número de secuencias a analizar y la matriz de distancias.

Para iniciar se debe indicar el número de OTUS a analizar en la casilla de sequence count y dar clic en New Input Size para indicar el tamaño de la matriz que se introducirá. Luego, proceder a introducir los datos en la parte editable. Debido a que la matriz es simétrica la aplicación automáticamente llena las casillas restantes con los datos proporcionados. El programa permite generar datos aleatorios para hacer pruebas del programa con el botón Random Data. Finalmente para obtener el resultado se debe dar clic en el botón Build Trees.

En la siguiente figura se muestra la forma de introducción de los datos para el caso general descrito.

Figura 12. Interfaz gráfica de la implementación de Sestoft.



- Phylip

El programa Phylip: Neighbor – Neighbor Joining and UPGMA methods (Felsenstein). Al igual que la implementación de Sestoft recibe como dato de entrada la matriz de distancia a través de un archivo con extensión .data. El archivo para el caso general está estructurado como sigue:

```
5 20

secuencia0 0 2.52777777778 5.8541871037 5.187503916 4.57708210856

secuencia1 2.5277777778 0 4.57708210856 6.57987116357 5.85341871037

secuencia2 5.8541871037 4.57708210856 0 2.0537037037 4.57708210856

secuencia3 5.187503916 6.57987116357 2.0537037037 0 3.00185185185

secuencia4 4.57708210856 5.85341871037 4.57708210856 3.00185185185 0
```

La primera línea del archivo contiene el número de secuencias a analizar y la longitud de las secuencias. En las siguientes líneas los primeros 10 caracteres de cada una representan el nombre de la especie. Y los siguientes cada fila de la matriz de distancia. Todo separado por espacios.

Figura 13. Interfaz gráfica de Phylip.

Los datos mínimos requeridos para obtener el resultado es un email y cargar con la opción examinar el archivo .data o bien, ingresar directamente la información del archivo en el área de texto. Esta implementación permite ejecutar sobre los datos proporcionados dos métodos de análisis filogenético: el UPGMA y el Neighbor Joining. Para elegir el método a emplear se debe activar el radioButton indicado y ejecutar la aplicación dando clic sobre el botón Run Neighbor.

5.2.2 Resultados obtenidos en cada aplicación. A continuación se describen los resultados obtenidos con las diferentes implementaciones.

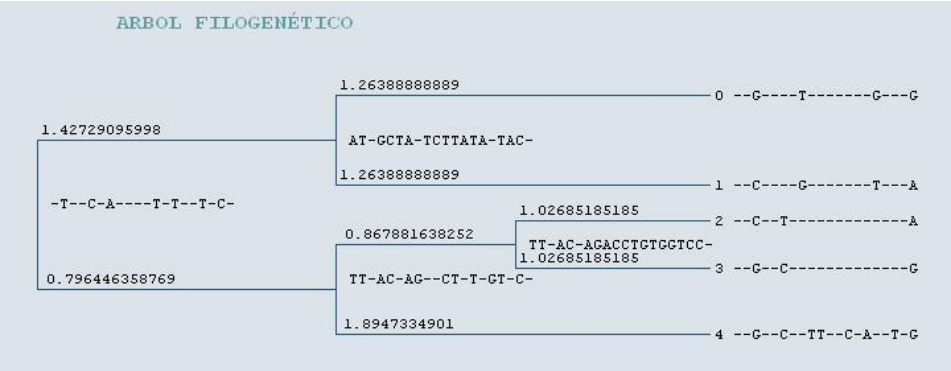
- UPGMA v1.0

Esta aplicación arroja como resultado dos archivos:

- Un archivo de texto con los pasos intermedios del algoritmo

- El árbol filogenético como se muestra en la siguiente figura:

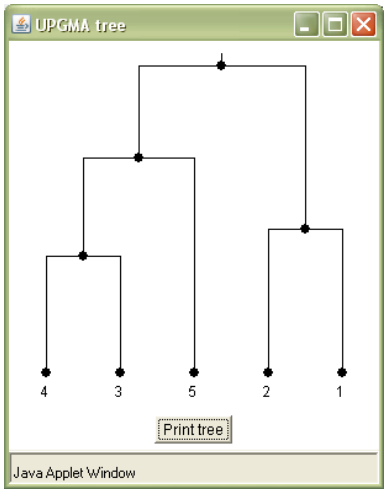
Figura 14. Resultado en UPGMA v 1.0



- Implementación de Sestoft

Esta aplicación arroja como resultado el árbol filogenético como se muestra en la siguiente figura:

Figura 15.Resultado de la implementación de Sestoft



- Phylip

Esta aplicación arroja los siguientes resultados:

- Outfile: contiene la matriz de distancias proporcionada y el árbol resultado en formato texto.

- Outtree: contiene el árbol resultado en formato de Newick. Y permite guardar un postscript del dibujo del árbol resultado.

Figura 16. Resultado Phylip - Postscript

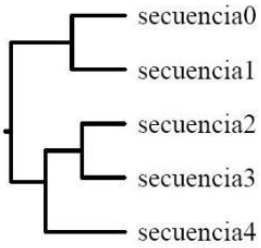
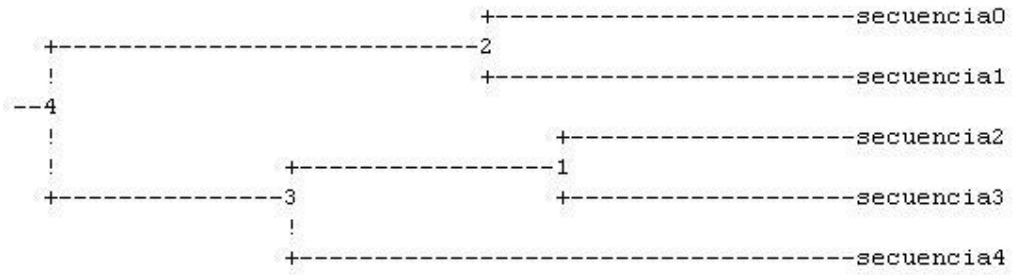


Figura 17. Resultado de Phylip – Archivo outfile



- Outtree indent: contiene el mismo árbol anterior indentado.

- Neighbor out: parámetros de las diferentes iteraciones

- Estándar error file: contiene una descripción de los errores que se produjeron en la ejecución.

5.2.3 Análisis de resultados. Los datos mostrados en el caso general fueron analizados con la aplicación UPGMA v1.0 y el resultado obtenido se mostró en la figura 14. Las mismas secuencias fueron analizadas con Phylip y con la implementación de Sestoft y los resultados obtenidos se muestran en las figuras 15 y 17 respectivamente. Se observa que los resultados son iguales.

En la aplicación UPGMA v1.0 propuesta en el presente proyecto el árbol presenta información adicional que no es incluida en las otras implementaciones. El fenograma incluye datos como las secciones de las secuencias que son similares en las OTUS agrupadas, distancia de las ramas y las secciones de las secuencias diferentes en cada especie analizada. Además de lo anterior, también en los resultados se incluye un archivo que contiene los pasos intermedios y los cambios realizados en la matriz de distancia hasta agrupar todas las secuencias en el árbol.

Cada una de las tres implementaciones cuyos resultados para el caso general fueron mostrados en la sección 5.2.2 poseen ventajas y desventajas en cuanto a la recepción de los datos de entrada y su resultado. A continuación se mencionan algunas de ellas.

- Phylip: entre las características a favor de esta implementación se pueden señalar las siguientes: en una implementación de código abierto y puede ser usada libremente. No necesariamente debe ser instalada en el equipo para ser utilizada. Ya que se puede acceder a ella a través de la Web. Tiene una interfaz gráfica sencilla. Proporciona los resultados en archivos que pueden ser guardados y abiertos posteriormente. Además genera un archivo con los errores que pudieron haberse cometido en la ejecución, lo que orienta al usuario en la utilización del

software. La principal desventaja del programa es que requiere como dato de entrada la matriz de distancia ya formada y no las secuencias. Estos datos se capturan a través de una archivo o digitados directamente en la interfaz de usuario.

- Implementación de Sestoft: esta aplicación no requiere ser instalada en el equipo ya que se puede utilizar en linea. Posee una interfaz de usuario sencilla. Al igual que Phylip solicita como dato de entrada la matriz de distancia ya formada, sin embargo, sólo acepta los datos digitados directamente o generación de datos aleatorios. Esta característica hace tediosa la digitación de los datos cuando se trata de análisis de un gran número de secuencias.

- UPGMA v1.0: A diferencia de las dos implementaciones descritas anteriormente esta requiere ser instalada en el equipo y funciona stand-alone. Posee una interfaz sencilla y con un entorno similar al de Windows que guía al usuario en los pasos a seguir para realizar el análisis. Requiere como dato de entrada un archivo con las secuencias a analizar y genera automáticamente la matriz de distancias evitándole al usuario esta tarea. Genera como resultados además del árbol un archivo con los cambios realizados en la matriz de distancia en los pasos intermedios.

5.3 TIEMPO DE EJECUCIÓN

Las pruebas para medir el tiempo de ejecución del programa se realizaron en un equipo con las siguientes características:

- Sistema:
  Microsoft Windows XP Profesional Versión 2002 SP2

- Equipo:

  Procesador AMD 1700+. 1.47GHz. 512 MB de RAM.

Las secuencias utilizadas en los archivos de prueba corresponden a datos generados de manera aleatoria.

La prueba realizada corresponde a la medición del tiempo de ejecución. Esto se logró tomando dos datos: un tiempo inicial cuando el programa recibe la instrucción de analizar las secuencias y un tiempo final después de que el programa dibuja el árbol resultado. El tiempo fue tomado en el siguiente formato:

Hora : Minutos : Segundos.m icrosegund os
Ej :10 : 02 : 29.750000

A continuación se presentan algunos de los datos tomados en las pruebas.

Tabla 8. Tabla de tiempo de ejecución con cantidad de secuencias constante.

Cantidad de Secuencias	Longitud de las secuencias	Tiempo de ejecución (en microsegundos)
5	5	31000
5	10	32000
5	15	32000
5	20	31000
5	50	31000
5	100	31000
5	125	32000
5	120	31000
5	170	31000

Tabla 9. Tabla de tiempo de ejecución con longitud de las secuencias constante.

Cantidad de Secuencias	Longitud de las secuencias	Tiempo de ejecución (en microsegundos)
4	5	31000
5	5	32000
6	5	32000
7	5	63000
8	5	172000
9	5	937000

Tabla 10. Tabla incrementando cantidad de secuencias y longitud de secuencias.

Cantidad Longitud	Tiempo de ejecución (en microsegundos)
4-4	31000
5-5	31000
6-6	72000
7-7	104000
8-8	354000
9-9	7011000
10-10	348735000

Con base en los datos presentados en las tablas anteriores se crearon las siguientes gráficas.

Figura 18. Longitud de las secuencias VS Tiempo de ejecución

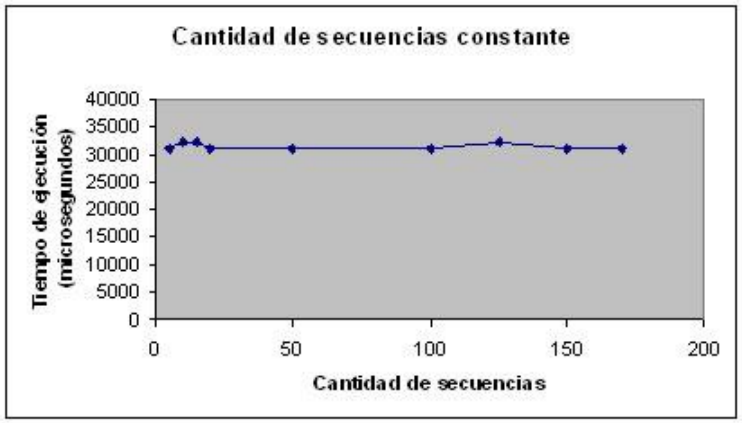


Figura 19. Cantidad de secuencias VS Tiempo de ejecución

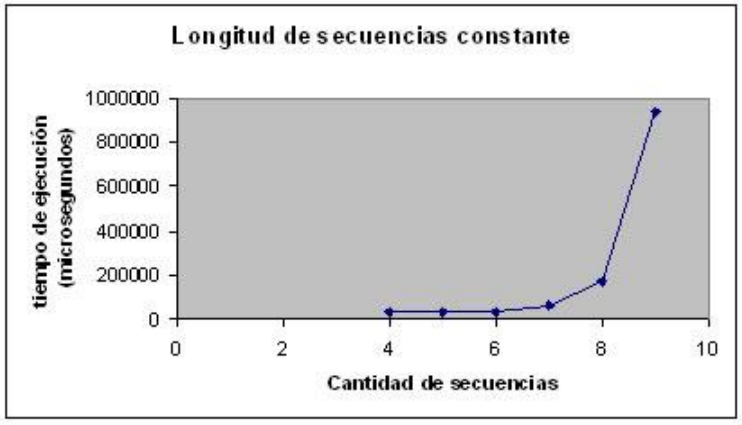
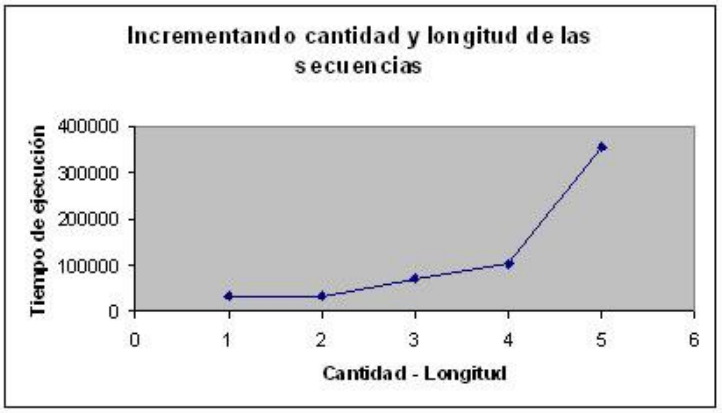


Figura 20. Cantidad – Longitud VS Tiempo de ejecución.



La figura 18 se creó con los datos mostrados en la tabla 8. Para tomar estos datos se usaron nueve archivos de prueba que contenían 5 secuencias cada uno. En cada archivo se emplearon secuencias con longitud diferente. Las longitudes de secuencia usadas fueron: 5, 10, 15, 20, 100, 125, 120 y 170. Las pruebas realizadas muestran que el programa funciona bien para secuencias con longitudes menores de 170. La gráfica de la figura 18 revela que el tiempo de ejecución del programa no depende de la longitud de las secuencias. Esto se puede afirmar debido a que aunque se varíe la longitud de las secuencias en las pruebas, el tiempo se mantuvo constante.

La figura 19 se creó con los datos mostrados en la tabla 9. Para tomar estos datos se usaron seis archivos de prueba que contenían secuencias de longitud 5 cada uno. En cada archivo se emplearon distinta cantidad de secuencias. Las cantidades de secuencia usadas fueron: 4, 5, 6, 7, 8 y 9. Las pruebas realizadas muestran que el programa funciona bien para archivos que contengan un número de secuencias menor o igual a 9. La gráfica de la figura 19 revela que el tiempo de ejecución del programa depende directamente de la cantidad de secuencias que contenga el archivo. Esto se puede afirmar debido a que cuando se varía la cantidad de secuencias, el tiempo de ejecución del programa crece exponencialmente.

En la figura 20 se muestra la gráfica creada con los datos de la tabal 10. Para tomar estos datos se usaron siete archivos de prueba que contenían cantidad de secuencias diferentes y de distinta longitud. Las cantidades de secuencia usadas fueron: 4, 5, 6, 7, 8, 9 y 10 y las longitudes de las secuencias fueron 4, 5, 6, 7, 8, 9 y 10, respectivamente. Las pruebas realizadas muestran que el programa funciona bien para archivos que contengan un número de secuencias y longitud de secuencias menores o iguales a 10. De la gráfica de la figura 20 comprueba la conclusión extraída de la gráfica anterior dado que muestra que el tiempo crece exponencialmente al aumentar la cantidad de secuencias.

CAPÍTULO 6: CONCLUSIONES

En el capítulo anterior se realizó una descripción de los resultados del trabajo de investigación. Se presentó una comparación de la implementación propuesta con otras dos implementaciones. Y también se muestra el análisis de las pruebas realizadas con el fin de determinar el comportamiento del software en relación al tiempo de ejecución. En este capítulo se presentarán las conclusiones finales de la realización de este proyecto.

Los objetivos definidos desde el inicio de la tesis se han cubierto ampliamente. En el primer capítulo de la tesis se enuncia como propósito principal el diseño e implementación de una interfaz de software para análisis filogenético basada en el algoritmo UPGMA. Al finalizar este proyecto se presenta un sistema de información que permite analizar una cantidad de 10 o menos secuencias de ADN de longitud menor o igual a 170 con el método UPGMA para tal fin.

A medida que avanzaba la investigación surgieron algunas problemáticas que se incorporaron como parte de este trabajo. Entre estas puede mencionarse la insuficiencia inicial de conocimientos biológicos y de las metodologías necesarias que incluye la filogenética como base principal del proyecto. Con el propósito de compensar esta insuficiencia se planteó el primer objetivo específico como recopilación y análisis de la información necesaria para establecer el marco teórico de la investigación. Este hecho quedó evidenciado en los capítulos iniciales de este trabajo.

El resultado de esta investigación está enmarcado dentro del área de la bioinformática. Una integración de la informática y la biología que establece una relación entre las teorías y métodos de la filogenética y las técnicas de la ingeniería del software y que se complementan en una implementación para análisis filogenético de secuencias de ADN.

BIBLIOGRAFÍA

ALLEN G, Rodrigo; GERALD H, Learn. Computacional and Evolutionary Analysis of HIV Molecular Sequences. New York : Kluwer Academic Publishers, 2000. 309 p. ISBN 0-7923-7994-2.

CECCHI, MARÍA CLAUDIA, GUERRERO-BOSAGNA, CARLOS y MPODOZIS, JORGE. El ¿delito? de Aristóteles. Rev. chil. hist. nat. [online]. set. 2001, Vol.74, no.3 [citado 30 Enero 2007], p.507-514. Disponible en Internet: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0716-078X2001000300001&lng=es&nrm=iso. ISSN 0716-078X.

Chris Bystroff. Bioinformatics. Fecha de consulta: Marzo 2 de 2007. Depts of Biology & Computer Science Rensselaer Polytechnic Institute. Disponible en: www.bioinfo.rpi.edu/~bystrc/courses/biol4540.html

COFFIN, John M.; HUGBERS, Stephen H.; VARMUS, Harold E. Retroviruses. New York:Cold Spring Harbor Laboratory Press. 1999. 843 p. ISBN:0-87-969571-4.

Cornell University Library. The PHYLogeny Inference Package. [Online]. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://vivo.library.cornell.edu/entity?home=1&id=5111

CRISP, Michael. Introductory glossary of cladistic terms. [Online]. Fuente: Invited Contributions of the Society of Australian Systematic Biologists. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.science.uts.edu.au/sasb/glossary.html

Emilio J. López Caballero y Gonzalo Pérez Suárez. Métodos de análisis en la reconstrucción filogenética. Departamento de Biología Animal. Universidad de Alcalá. Alcalá de Henares. Madrid. España. nº 26, [citado 2 de febrero de 2007] 1999 : 45-56. Disponible en Internet: http://entomologia.rediris.es/sea/bol/vol26/s1/articulo/index.htm

Dave Thomas, NMSR. Example calculation of phylogenies: The UPGMA method. [Online]. Fecha de consulta: 5 de febrero de 2007. Rosswell, New Mexico. Última modificación: 31 de octubre de 2002. Disponible en: http://www.nmsr.org/upgma.htm

DAWSON, Michael. Python Programming for the Absolute Beginner. Boston: Course PTR. 2003. 480 p. ISBN 1-59200-073-8.

DEPARTMENT OF BIOLOGY. SOUTHERN OREGON UNIVERSITY. Compleat Cladist: A primer of phylogenetic procedures. [Online]. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.sou.edu/biology/Courses/Bi332/CompleatCladist.pdf

Dr. Steven M. Carr., Cluster Analysis: an example. [Online]. Fecha de consulta: 5 de febrero de 2007. Memorial University of Newfoundland. Genetics, Evolution, and Molecular Systematics Laboratory. Department of Biology. St. John's NF A1B 3X9, Canada. Disponible en: http://www.mun.ca/biology/scarr/Bio4900_UPGMA.html

FACULTAD DE CIENCIAS UNIVERSIDAD AUTÒNOMA DE MEJICO. Index of /Bioinformática. [ONLINE]. Fecha de consulta: 22 agosto 2006. Disponible en internet: http://bacteria.fciencias.unam.mx/Bioinformatica

FELSENSTEIN, Joe. Phylip. [Online]. Fecha de consulta: 28 agosto 2006. Disponible en Internet: http://evolution.genetics.washington.edu/phylip.html

FELSENSTEIN, Joe. Phylogeny programs. [Online]. Fecha de consulta: 28 agosto 2006. Disponible en Internet: http://evolution.genetics.washington.edu/phylip/software.html

GRIBSKOV, M. Sequence Análisis Primer. New York: Oxford University Press, Incorporated. 1991. 296 p. ISBN 0-19-509874-9.

GONZALEZ, G. Los Coccinellidae de Chile. [Online]. 1996. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.coccinellidae.cl

Günter Bechly (Böblingen, Alemanha). Glossary of Phylogenetic Systematics. [Online]. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.bernstein.naturkundemuseum-bw.de/odonata/glossary.htm

JUKES, T.H., y C.R. CANTOR. 1969. Evolution of Proteins Molecules. Pp 21-132 en H:N: MUNRO, ed. Mammalian protein metabolism. Academis Press, New York.

KUHNER MK; FELSENSTEIN J. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. [Online]. Volumen 12, Número 3. Mayo de 1995. Disponible en Internet: http://mbe.oxfordjournals.org/archive/

Leitner T, Escanilla D, Franzen C, Uhlen M, Albert J. Accurate reconstruction of a known HIV-1 transmission history by phylogenetic tree analysis. 1996 Oct 1 93:10864-9. Disponible en Internet: www.pubmed.com

Los Alamos National Laboratory. HIV sequence Database. [Online]. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://hiv-web.lanl.gov/

Los Alamos National Laboratory. HIV and SIV Nomenclature. [Online]. Fecha de consulta: 22 agosto 2006. Última modificación: Mon Apr 24 19:23 2006. Disponible en: http://hiv-web.lanl.gov/content/hiv-db/HelpDocs/subtypes-more.html.

MacLEOD, Norman; FOREY, Peter L. Morphology, Shape & Phylogeny. London: Taylor & Francis, Incorporated. 2002. 318 p. ISBN 0-203-16517-9.

MIYAMOTO, Michael M. ; CRACRAFT, Joel. Phylogenetic Analysis of DNA Sequences. New York: Oxford University Press, Incorporated. 1991. 369 p. ISBN 0-19-506698-7.

MOTOO Kimura, TOMOKO Ohta. On the stochastic model for estimation of mutational distance between homologous proteins. Journal of Molecular Evolution. [Online]. Volume 2. Issue 1. Monday, May 16, 2005. [Citado marzo 8 2008]. Disponible en http://www.springerlink.com/content/u244547185w7306w. ISSN 1432-1432 , Pages 87-90.

MOUNT, David W. Bioinformatics: Sequence and Genome Analysis. New York: Cold Spring Harbor Laboratory Press. 2001. 577 p. ISBN 0-87969-597-8.

MURPHY, Robert F. Introduction to Computacional Molecular Biology. [Online]. 2006. Departments of Biological Sciences and Biomedical Engineering Carnegie Mellon University. Última modificación enero 6 2006. Disponible en Internet: http://www.cmu.edu/bio/education/courses/03311/

NING K., SHAN T., XIANG S. L., SHEN W. Phylogenetic Tree Reconstruction: Distance Based. [Online]. Octubre 10 de 2003. Disponible en: http://www.comp.nus.edu.sg/~bioinfo/phylogenetic%20tree%20reconstruction_2_8.pdf . Fecha de consulta: 13 de febrero de 2007. 20 páginas.

OMS/ONUSIDA. Los asociados mundiales en pro de una vacuna contra el VIH fortalecen su colaboración para acelerar los avances. [Online]. 7 de febrero de 2005. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.who.int/entity/mediacentre/news/notes/2005/np04/es/index.html

OOH SING Hua, OOI HONG Sain, WONG CHEE Hong, WONG SUM Thai. Phylogenetics Trees Reconstruction. [Online]. Septiembre 26 de 2003. Disponible en: http://www.comp.nus.edu.sg/~bioinfo/phylogenetic%20tree%20reconstruction_1_7.pdf . Fecha de consulta: 28 de marzo de 2007. 26 páginas.

Python Software Foundation. About Python. [Online]. Fecha de consulta: 28 agosto 2006. Disponible en internet: http://www.python.org/about/

Python Software Foundation. FAQ General de Python. [Online]. 16 de diciembre del 2005. Fecha de consulta: 28 agosto 2006. Disponible en Internet: http://www.python.org/doc/faq/es/general/#faq-general-de-python
RILEY, Sean. Game Programming with Python . Hingham, Massachusetts: Charles River Media. 2003. 484 p. ISBN 1-58450-258-4.

SAKSENA NK, WANG B, Ge YC, XIANG SH, DWYER, DE , CUNNINGHAM AL. Coinfection and genetic Recombination between HIV-1 strains: possible biological implications in Australia and South East Asia. 26:121-7. Disponible en Internet: www.pubmed.com

SIDDALL, Mark E. Phylogentics: Just Methods. [Online]. Fecha de consulta: 28 agosto 2006.American Museum of Natural History. Disponible en internet: http://research.amnh.org/~siddall/methods/

SORENSEN, Daniel; GIANOLA, Daniel. Likelihood, Bayesian and MCMC Methods in Genetics. New York: Springer-Verlag New York, Incorporated. 2002. 757 p. ISBN 0-387-22764-4.

TAJIMA, Fumio. MSATOCHI, Nei. Unbiased Estimation of Evolutionary Distance between Nucleotide Sequences. Department of population Genetics, Nacional Institute of genetics. University of Chicago. Mol. Biol. Evol. 10(3). Pp. 677-688. Octubre 6 de 1992.

UNIVERSIDAD DE CALIFORNIA, MUSEUM OF PALEONTOLOGY. Journey into the phylogenetics Systematics. [Online]. Berkeley, EUA. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.ucmp.berkeley.edu/clad/clad4.html

UNIVERSIDAD DE CALIFORNIA, MUSEUM OF PALEONTOLOGY. The Phylogeny of life. [Online]. Berkeley, EUA. Fecha de consulta: 22 agosto 2006. Disponible en Internet: http://www.ucmp.berkeley.edu/alllife/threedomains.html

UNIVERSIDAD DE LA REPUBLICA. Sistemática Biológica: Métodos Y Principios. Fecha de Consulta: 1 de marzo de 2007. Facultad de ciencias. Laboratorio de

evolución.Montevideo, Uruguay. Disponible en: http://evolucion.fcien.edu.uy/sistematica/intro-cladistica.pdf

WANG, Jason T. L.; WU, Cathy H.; WANG, Paul P. Computacional Biology and Genome Informatics. Londres : Scientific Publishing Company, Incorporated. 2003. 266 p. ISBN 9-81-238257-7.

ANEXOS

ANEXO A
Ejemplo de archivo de entrada

>001 secuencia de prueba no real|len=8
AGTAGTTC

>002 secuencia de prueba no real
AGTAGTTA

>003 secuencia de prueba no real
AGTAGTAA

>004 secuencia de prueba no real
AGTAGGGG

>005 secuencia de prueba no real
AGTAGGGC

ANEXO B
Ejemplo de archivo de salida (a)

Pasos Intermedios

Ejecutando UPGMA

FECHA: 2007-6-8

HORA: 17:52:45

Archivo creado con UPGMA v1.0

SECUENCIAS

AGTAGTTC

AGTAGTTA

AGTAGTAA

AGTAGGGG

AGTAGGGC

MATRIZ DE DISTANCIAS

0	0.875	1.375	1.74537037037	1.375
0.875	0	0.875	1.74537037037	1.74537037037
1.375	0.875	0	1.74537037037	1.74537037037
1.74537037037	1.74537037037	1.74537037037	0	0.875
1.375	1.74537037037	1.74537037037	0.875	0

Inicio

PASO1

Menor distancia 0.875

NUEVA MATRIZ

0	0.125	1.74537037037	1.56018518519
1.125	0	1.74537037037	1.74537037037
1.74537037037	1.74537037037	0	0.875
1.56018518519	1.74537037037	0.875	0

ARBOL[[0, 1, 0.4375, 'AGTAGTT-', 0.4375, 0.4375], 2, 3, 4]

PASO2

Menor distancia 0.875

NUEVA MATRIZ

0	1.125	1.65277777778
1.125	0	1.74537037037
1.65277777778	1.74537037037	0

ARBOL[[0, 1, 0.4375, 'AGTAGTT-', 0.4375, 0.4375], 2, [3, 4, 0.4375, 'AGTAGGG-', 0.4375, 0.4375]]

PASO3

Menor distancia 1.125

NUEVA MATRIZ

0	1.69907407407
1.69907407407	0

ARBOL[[[0, 1, 0.4375, 'AGTAGTT-', 0.4375, 0.4375], 2, 0.5625, 'AGTAGT--', 0.125, 0.5625], [3, 4, 0.4375, 'AGTAGGG-', 0.4375, 0.4375]]

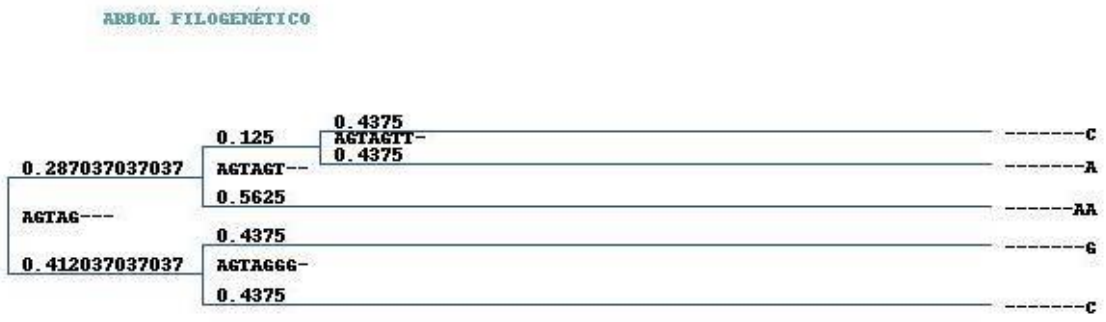
PASO4

Menor distancia 1.69907407407

NUEVA MATRIZ
0

ARBOL[[[0, 1, 0.4375, 'AGTAGTT-', 0.4375, 0.4375], 2, 0.5625, 'AGTAGT--', 0.125, 0.5625], [3, 4, 0.4375, 'AGTAGGG-', 0.4375, 0.4375], 0.84953703703703698, 'AGTAG---', 0.28703703703703698, 0.41203703703703698]

ANEXO C
Ejemplo de archivo de salida (b)



ANEXO D
Formato de archivo de entrada para Phylip

Archivo : infile.data

[Número de secuencias] [Longitud de las secuencias]

[Id1] [matDistancia[0][0] … matDistancia[0][j] … matDistancia[0][n-1]]

…

[Idi] [matDistancia[i][0] … matDistancia[i][j] … matDistancia[i][n-1]]

…

[Idn] [matDistancia[n-1][0] … matDistancia[n-1][j] … matDistancia[n-1][n-1]]

Convenciones:

Idk = identificador de la secuencia. Comprende los 10 primeros caracteres de cada línea

matDistancia[i][j]= valor correspondiente a la posición (i,j) de la matriz de distancia

ANEXO E
Ejemplo de archivo de entrada para Phylip

Archivo : infile.data

5 20

secuencia0 0 2.52777777778 5.8541871037 5.187503916 4.57708210856

secuencia1 2.5277777778 0 4.57708210856 6.57987116357 5.85341871037

secuencia2 5.8541871037 4.57708210856 0 2.0537037037 4.57708210856

secuencia3 5.187503916 6.57987116357 2.0537037037 0 3.00185185185

secuencia4 4.57708210856 5.85341871037 4.57708210856 3.00185185185 0