Universitat Autònoma de Barcelona

escola d'enginyeria

INGENIERÍA INFORMÁTICA

2026 BIOINFORMÁTICA:

BÚSQUEDA DE ANCESTROS COMUNES ENTRE GENOMAS DE DIFERENTES ESPECIES

Memoria del Proyecto Final de Carrera
de Ingeniería Informática
realizado por
Jonas Rodriguez Murillo
y dirigido por
Jordi Gonzàlez Sabaté
Bellaterra, 9 de Septiembre de 2010

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Jordi Gonzàlez Sabaté

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Jonas Rodriguez Murillo.

I per tal que consti firma la present.

Signat: .....

Bellaterra, ...de.....de 2010

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Mario Huerta

de l'Institud de Biomedicina i Biotecnologia de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Jonas Rodriguez Murillo.

I per tal que consti firma la present.

Signat: .....

Bellaterra, ...de.....de 2010

Agradecimientos

En primer lugar me gustaría agradecer toda la colaboración y la confianza depositada en mi trabajo por mi tutor del proyecto, Mario Huerta. Sin su ayuda este proyecto no hubiera sido posible.

También me gustaría dar las gracias a mi familia y a Sandra Sumoy por su paciencia y comprensión durante todos los meses que ha durado el proyecto.

Índice

1. Introducción

1.1 Motivación Proyecto

La comparación de genomas proporciona mucha información sobre los procesos evolutivos que han llevado a la aparición de los diferentes seres vivos que pueblan el planeta tierra.

Pero además, esta comparación de genomas, tiene una muy importante aplicación médica. Actualmente la asignación de función a los genes suele hacerse por reconocimiento de subsecuencias funcionales conservadas de un organismo a otro.

Personalmente escogí este proyecto porque considero que la colaboración en un proyecto de investigación sobre genómica supone un atractivo reto desde el punto de vista formativo. Además, el diseño e implementación de los diferentes algoritmos para la realización de los programas será un desafío muy interesante donde poner a prueba todos los conocimientos que he adquirido durante los años de carrera.

1.2 Estado del Arte

Podemos definir la genética comparativa como el estudio de las semejanzas y diferencias entre genomas de diferentes organismos en un intento de entender los procesos evolutivos que actúan sobre dichos genomas.

Para comparar diferentes genomas se buscan similitudes entre sus secuencias, es decir, determinadas subsecuencias compartidas por ambos genomas. Una de las posibles estrategias para dicha comparación es el uso de *Maximal Unique Matchings* (MUMs). Los MUMs serían las subsecuencias comunes más largas y únicas, es decir no repetidas en el resto de las secuencias comparadas.

Existen diversas herramientas para la comparación entre genomas, cabe mencionar algunas basadas en la estrategia de MUMs que están relacionadas con el proyecto:

- MUMmer (1999). Aplicación para la alineación de genomas completos. Se basa en la búsqueda de MUMs. Solo puede realizar la comparación de dos genomas simultáneamente. Para ello se construye un único *suffix-tree* o árbol de sufijos que contiene los sufijos de ambos genomas, donde las ramas más largas son los MUMs coincidentes entre genomas. [1]

- MUMs On-Line (2002). Evolución del cálculo de los MUMs a partir del algoritmo suffix-trees. Introduce una modificación para reducir el espacio de construcción de éstos, que consiste en crear un único suffix-tree que contendrá sólo los sufijos de un

genoma y que una vez creado podrá usarse para comparar dicho genoma con el resto de genomas. Esto implica que el espacio a ser utilizado no es lineal respecto a la longitud de todos los genomas como venía siendo, sino a la del genoma más pequeño. El algoritmo MUMs On-line es posible gracias a una variación en el algoritmo de construcción de los suffix-trees y en el suffix-tree generado. Esta variación recibe el nombre de *slide suffix-trees*. [2] [4]

- MALGEN (2003). Es el acrónimo de *Multiple ALignment of GENomes* y es una herramienta para la exploración de relaciones entre secuencias de ADN. Está basada en el cálculo de MUMS y pueden ser calculadas y representadas por más de dos secuencias simultáneamente, haciendo uso del algoritmo de MUMs On-Line para encontrar los MUMs. Esta herramienta es accesible vía web. [3]

- M-GCAT (2006). Es el acrónimo de Multiple Genome Comparison and Alignment Tool y es una herramienta interactiva para la comparación de genomas. Puede realizar comparaciones de varios genomas simultáneamente. Se basa en *slide suffix-trees* añadiendo ciertas herramientas de post proceso:
    - Anchoring: Para poder alinear eficientemente todos los genomas es necesario limitar el espacio de programación dinámica a través de la búsqueda heurística. Este anclaje es un método heurístico que puede ser usado para establecer un marco de secuencia conservada entre todas las secuencias que se comparan.
    - Recursive anchoring: El objetivo es recorrer los genomas con la mayor concordancia posible por medio de búsquedas en las regiones que se encuentran entre las anclas para la creación de nuevas regiones lo suficientemente pequeñas como para ser alineados de manera eficiente.
    - Filtering: El objetivo de este paso es la eliminación de ruido.
    - Clustering: Organización y agrupamiento de las regiones conservadas en los pasos previos.

  Esta aplicación no está orientada a web, sino que es accesible para descarga y ejecución local. Con ella se obtienen los cálculos de MUMs con una mayor eficiencia temporal. [4]

Si se quieren conocer los ancestros comunes de diferentes especies no es suficiente con saber que partes del genoma son idénticas, sino que hay que buscar que partes se han modificado, si alguna se ha dividido, etc. Información que con solo los MUMs no se puede obtener, dado que como se trata de *exact matchings* nos da información sobre lo que se ha conservado pero no sobre lo que se ha modificado pero dentro de una región altamente conservada. Para ello necesitaremos *aproxímate string matchings*.

Se necesita una unidad de comparación algo más flexible que los MUMs, que permita observar las similitudes y diferencias entre las regiones de diferentes genomas. Y que además ayude a reducir el coste de los cálculos con MUMs.

Estas nuevas unidades de matching serán superestructuras que contendrán agrupaciones de MUMs ya creado y los espacios (*gaps*) que hay entre ellos. De esta manera pasaremos de un grado de similitud exacto entre dos regiones (*Exact Matchings)* a comparar cadenas aproximadas (*Aproximate String Matching)* puesto que los *gaps* pueden variar de un genoma a otro aunque la región y parte codificante de los genes se conserve (con lo que el gen se está conservando). Estas nuevas superestructuras recibirán el nombre de superMUMs.

Para poder realizar los programas del proyecto correctamente, se deberá resolver el principal problema de trabajar con MUMs: la cantidad enorme de MUMs de algunas comparaciones. Para hacernos una idea de ello la comparación de los genomas de especies del dominio[1]de Eucariotas pueden pasar de los 83 Millones de MUMs, esto implica que el diseño de cualquier programa que trabaje con MUMs deba ser muy eficiente si quiere tener tiempos viables de cómputo.

La búsqueda de los ancestros comunes que se llevará a cabo en la segunda parte del proyecto servirá para el estudio de la conservación regiones conservadas de una especie a otra, la asignación de propiedades de un gen a otro y el estudio de las variaciones producidas durante la evolución de las especies.

La base de la búsqueda de ancestros comunes es la conservación de regiones de diferentes genomas. Y para la búsqueda de las regiones conservadas se utilizaran los nuevos superMUMs. Serán muy útiles en este caso ya que gracias a que los MUMs que forman un superMUM son únicos solo se tiene que buscar en ese punto del otro genoma.

1.3 Objetivos

El propósito de este proyecto es la creación de una serie de métodos y algoritmos que permitan encontrar las relaciones entre diferentes especies a partir de las similitudes de sus genomas obteniendo así los ancestros comunes.

Los objetivos a realizar para llevar a término este proyecto serán los siguientes:

1. La creación de un algoritmo capaz de generar la nueva estructura de superMUMs basándose en los ficheros de MUMs de la comparación entre dos genomas. Y la implementación de un código de automatización para que se pueda utilizar el algoritmo de los superMUMs.

   1.1 Deberá ser diseñada como un pre-proceso. El programa solo se ejecutara una vez por cada pareja de genomas a comparar y se almacenaran los superMUMs resultantes en el servidor[6].

   1.2 La optimización del cálculo de superMUMs para mejorar su eficiencia. Es importante que la generación de superMUMs sea correcta pero también rápida, ya

[1] Los diferentes dominios de especies están descritos en el siguiente apartado fundamentos teóricos.

que en algunos casos la cantidad de MUMs que debe tratar es tan grande que sin optimización el tiempo de cálculo no seria viable.

2. La creación de un algoritmo para la búsqueda de regiones conservadas de un genoma a otro. La implementación del código para realizar la búsqueda de regiones entre todas las especies. Cuando esté finalizada será una herramienta on-line del servidor web [6] donde el usuario introduce el genoma y la región de la que quiere buscar su conservación en el resto de genomas.

   2.1 La creación de un algoritmo para el cálculo de la similitud entre las regiones encontradas formando una matriz de similitudes.

   2.2 La construcción de un árbol filogenético a partir de la matriz de similitudes.

1.4 Organización de la Memoria

Las aplicaciones desarrolladas para el pre-proceso tienen la finalidad de generar superMUMs de la comparación entre dos genomas utilizando para ello los ficheros de MUMs de esas comparaciones. También deberán automatizar el proceso, es decir calcular los superMUMs de todos los ficheros MUMs que estén dentro de una carpeta en concreto.

Las aplicaciones desarrolladas para el trabajo online servirán para buscar la conservación de regiones de un genoma en el resto de especies a partir del genoma y región entrada por parámetro. Utilizaran el sistema de superMUMs

Para poder documentar todas las aplicaciones correctamente, la memoria del proyecto está dividida en ocho partes, siendo la introducción la primera. El resto de partes de la memoria son:

- En la segunda parte se explicaran los fundamentos teóricos básicos que el lector deberá conocer para poder situarse en el contexto correcto para entender el proyecto.
- En la tercera será el planteamiento que se ha desarrollado para cumplir los objetivos.
- En la cuarta se expondrán los resultados obtenidos a durante el desarrollo de las fases.
- En la quinta se detallara el software utilizado en el proyecto. Llamadas de los programas, formato de los ficheros, estructuras de archivos, etc.
- La sexta parte serán las conclusiones e impresiones personales sobre el proyecto.
- En la séptima parte estarán las referencias hechas durante la memoria
- Por último, en la octava parte se encontrara un resumen del proyecto.

Durante la realización del proyecto utilizaremos diferentes tipos de datos. Para los juegos de pruebas casi todos serán casos de estudio preparados. Pero cuando los algoritmos sean funcionales se incorporaran los cálculos con datos reales.

Como datos reales utilizaremos los genomas de los tres dominios de especies diferentes, Archaeas, Bacterias y Eucariotas[2]. Debido a la gran cantidad de MUMs que se generan en las comparaciones de Eucariotas, dejaremos los cálculos de estas para el final, cuando los algoritmos sean completamente funcionales y eficientes.

En el servidor del IBB [6] donde se realizara el proyecto ya están descargados los genomas de estas especies. Para las Archaeas tendremos los genomas de 48 especies para comparar y para Bacterias tendremos un subgrupo de 53. Aunque el número de especies de Archaeas y Bacterias a comparar es muy similar los genomas de estas últimas son significativamente más grandes, esto es patente cuando se han calculado comparaciones y se generan MUMs.

El número de MUMs con el que se trabajara puede variar mucho dependiendo con que especies se haga la comparación. En el caso de las Archaeas la comparación que consigue más MUMs tiene aproximadamente 18500 MUMs en el fichero. En las Bacterias ese número sube hasta aproximadamente 50 mil MUMs. Aunque ninguna de las dos tiene ni punto de comparación con los más de 83 millones que pueden alcanzar las Eucariotas.

[2] Ver la siguiente sección, fundamentos teoricos

2. Fundamentos Teóricos

2.1 Introducción Biológica

La Genética es el estudio de la naturaleza, organización, función, expresión, transmisión y evolución de la información genética codificada de los organismos. Se puede dividir en las siguientes áreas:

- Genética clásica (transmisión y localización de los genes en los cromosomas[3])
- Genética molecular (estructura y el control de la expresión del material genético)
- Genética evolutiva (referente a los procesos evolutivos de poblaciones)
- Genómica (correspondiente al análisis e interpretación de los genomas)

Este proyecto está orientado a la genómica comparativa que estudia las relaciones entre genomas de diferentes especies o razas. El objetivo que se busca es el de beneficiarse de la información proporcionada por las firmas de la selección natural para entender la función y los procesos evolutivos que actúan sobre los genomas.

Las secuencias de ADN que constituyen la unidad básica y funcional de la herencia se denominan genes[4], los cuales contienen la información genética de un genoma. Al conjunto de toda la información correspondiente a un organismo lo denominamos genotipo.

Cada célula de un organismo contiene el genoma completo, la diferencia que encontramos entre dos células distintas es que algunos genes están activos y otros no.

Podemos distinguir 3 dominios para clasificar las células según un modelo evolutivo de clasificación, propuesto por Carl Woese. Dicho modelo está basado en las diferencias encontradas en las secuencias de nucleótidos en los ribosomas[5] y RNAs[6] de transferencia de la célula, la estructura de los lípidos de la membrana, y la sensibilidad a los antibióticos.

Estos tres dominios son:

[3] Filamento condensado de ácido desoxirribonucleico, visible en el núcleo de las células durante la mitosis.
[4] Gen, proviene de la palabra griega γένος y significa "raza, generación".
[5] Orgánulo en el que tienen lugar las últimas etapas de la síntesis de proteínas.
[6] Ácido ribonucleico.

Tipo	Reino	Descripción
Bacteria (Eubacteria)	mycoplasmas cyanobacteria bacterias Gram-positivas bacterias Gram-negativas	▪ Son procariotas, contiene el ADN en el citoplasma. ▪ Son organismos microscópicos y en su mayor parte unicelulares. ▪ Son sensibles a los antibióticos antibacterianos tradicionales.
Archaea (Archaebacteria)	metanógeno halófilos extremos termoacidófilos	▪ Son procariotas, contiene el ADN en el citoplasma. ▪ Son organismos unicelulares. ▪ No son sensibles a algunos antibióticos que afectan a las Bacterias. ▪ Viven a menudo en ambientes extremos.
Eucariota (Eukaryota)	Hongos Protistas Plantas Animales	▪ Tienen un núcleo celular diferenciable que contiene el ADN. ▪ No son sensibles a los antibióticos antibacterianos tradicionales.

Podemos resumir que el dominio eucariota lo conforman plantas, hongos y animales, donde situamos a los mamíferos y por lo tanto al ser humano. Mientras que los dominios de Bacteria y Archaea los conforman respectivamente bacterias y arcobacterias.

Desde el punto de vista químico, el ADN es un polímero[7] de nucleótidos y cada nucleótido, a su vez, está formado por un azúcar (la desoxirribosa), una base nitrogenada (que puede ser adenina→A, timina→T, citosina→C o guanina→G) y un grupo fosfato que actúa como enganche de cada uno con el siguiente.

Lo que distingue un nucleótido de otro es la base nitrogenada, y por ello la secuencia del ADN se especifica nombrando sólo la secuencia de sus bases. La disposición secuencial de estas cuatro bases a lo largo de la cadena es la que codifica la información genética: por ejemplo, una secuencia de ADN puede ser ATGCTAGATCGC...

En los organismos vivos, el ADN se presenta como una doble cadena de nucleótidos, en la que las dos hebras están unidas entre sí por unas conexiones denominadas puentes de hidrógeno.

Simplificando, podemos decir que el ADN es un almacén cuyo contenido es la información de un organismo. Este conjunto de información se denomina genoma, el cual se compone de una cadena de genes, y que un gen es una sub-secuencia lineal de nucleótidos de la secuencia completa que compone un genoma.

[7] Un polímero es un compuesto formado por muchas unidades simples conectadas entre sí.

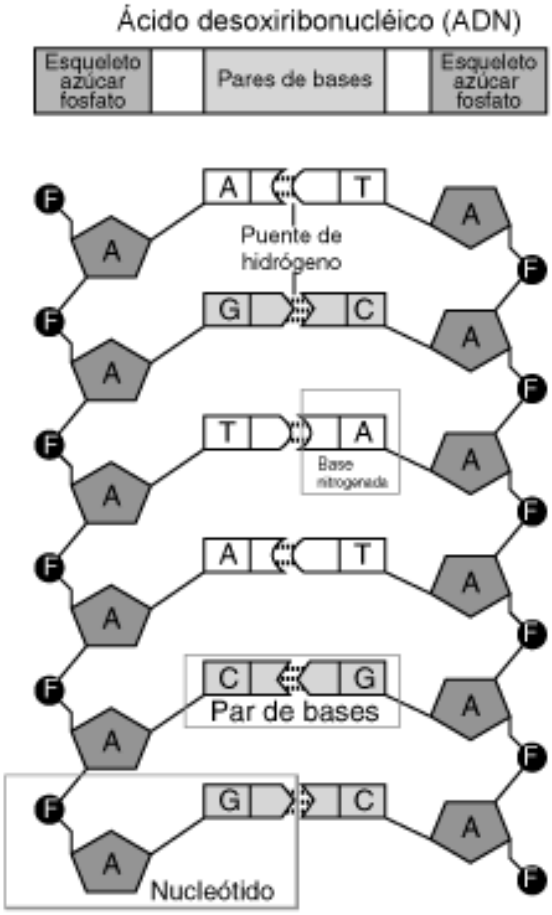


Figura 1 Estructura ADN

2.2 Arboles Filogenéticos

Un árbol filogenético es una clasificación científica que muestra las relaciones evolutivas entre varias especies u otras entidades que se cree que tienen una ascendencia común. Se representa con un cladograma, un diagrama de clasificación biológica de los organismos, en el que se muestra la relación entre distintas especies según una característica derivada.

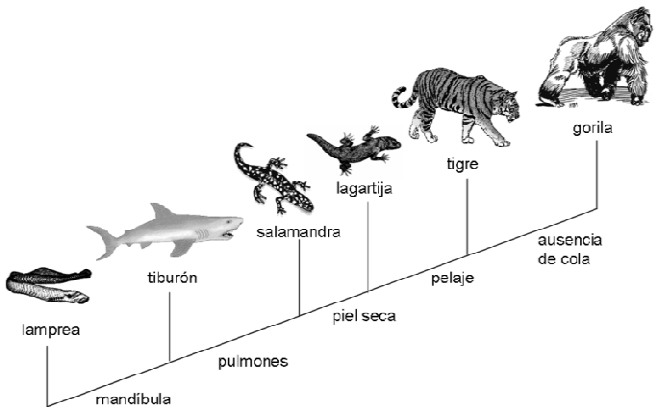


Figura 2 Ejemplo de cladograma simple de especies con características derivadas

El árbol filogenético es utilizado en biología para representar cómo se encuentran emparentados los organismos vivos. Para su construcción se usa información proveniente de fósiles, así como aquélla generada por la comparación estructural y molecular de los organismos. Los arboles filogenéticos tienen un tronco y ramas, en donde se muestran las relación entre especies.

Los árboles filogenéticos se construyen tomando en cuenta la teoría de la evolución, que nos indica que todos los organismos son descendientes de un ancestro común: la protocélula. Así, todos los organismos, ya sean vivos o extintos, se encuentran emparentados en algún grado.

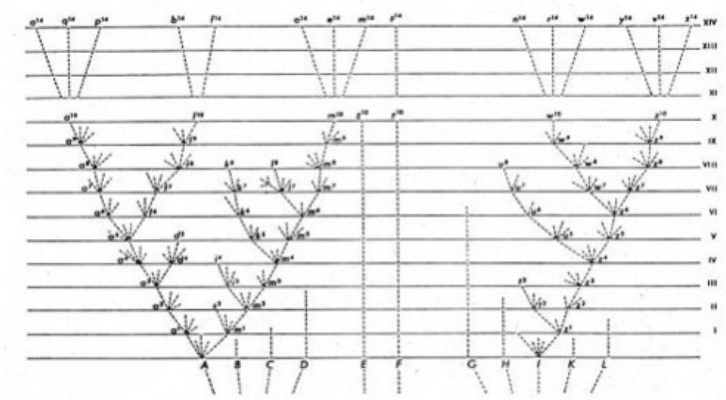


Figura 3 Diagrama dibujado por Charles Darwin en El Origen de las Especies

2.3 Ancestros Comunes

Buscar Ancestros comunes es lo mismo que buscar regiones conservadas de un genoma a otro. Si la región se conserva es que ambos genomas tienen un ancestro común del que heredaron esa región o bien un genoma es ancestro del otro.

La búsqueda de similitudes entre regiones de diferentes genomas aporta mucha información sobre las relaciones entre las especies de estos genomas como el estudio de la conservación de genes de una especie a otra, de cómo las propiedades de un gen son asignados a otro gen o de cómo se crean variaciones en genomas diferentes durante la evolución de esas especies.

2.4 Bioinformática y Genómica Comparativa

La bioinformática consiste en la gestión y el análisis de datos biológicos haciendo uso de la computación y tecnologías a nuestra disposición, abarcando los ámbitos de la matemática aplicada, la estadística, la inteligencia artificial, la química, la bioquímica, la informática o las ciencias de la computación. Podemos decir que es la ciencia dedicada al estudio de los fenómenos biológicos de la microbiología molecular desde un ámbito computacional, cuyo objeto es conseguir métodos robustos que faciliten la comprensión,

simulación y predicción de comportamientos biológicos observados en los seres vivos, mediante la utilización de recursos computacionales para buscar soluciones, para el análisis de datos o la simulación de sistemas y mecanismos de índole biológica (habitualmente a nivel molecular).

La bioinformática, unida al avance de las técnicas de manipulación del ADN, ha permitido determinar la secuencia completa del genoma de un organismo. De hecho parte de la importancia que tiene actualmente se debe al nacimiento del Proyecto Genoma Humano (HGP).

La secuencia que obtenemos de cada genoma aporta gran cantidad de información biológica de interés ya que permite predecir y catalogar el número total de genes y su estructura, así como definir la organización básica del organismo y las diferentes clases funcionales de proteínas (energía, comunicación, información, etc..).

Un ejemplo de la utilidad de los ámbitos en los que juega un papel fundamental, es el descubrimiento de nuevos fármacos. Además, combinada con las nuevas técnicas de biotecnología, la bioinformática permite identificar genes y proteínas causantes de enfermedades o de mecanismos de resistencia a antibióticos, que de otra forma serían complicados de detectar, permitiendo intervenir en los procesos con fármacos más específicos o, incluso, introduciendo nuevos paradigmas terapéuticos, como la terapia génica.

En relación con este proyecto, la comparación de secuencias entre diferentes especies resulta de gran importancia para comprender la evolución de éstas así como para entender la organización y funcionalidad de ciertos genes comunes entre dos especies distintas. Simplificando de esta manera la búsqueda de las regiones comunes que comparten especies con ancestros comunes.

La comparación de secuencias entre diferentes especies resulta de gran importancia para comprender la evolución de éstas así como para entender la organización y funcionalidad de ciertos genes comunes entre dos especies distintas. Simplificando de esta manera la búsqueda de las regiones que comparten especies con sus ancestros. Cuando se busca si áreas específicas de un genoma está conservada en algún punto de otro genoma, se pretende identificar un ancestro común para esta subsecuencia concreta.

2.5 Maximal Unique Matchings (MUMs)

Un método para realizar la tarea de comparación de dos secuencias genómicas distintas es buscar la sub-secuencia única más larga coincidente en ambos genomas, que debe aparecer una sola vez a lo largo la secuencia completa que conforma cada uno de los genomas. Estas sub-secuencias reciben el nombre de Maximal Unique Matchings (MUMs) y

el conjunto de MUMs obtenidos en una comparación determina el skeleton[8] sobre el cual se puede hacer una comparación global.

La búsqueda clásica de MUMs se realiza mediante la construcción de *suffix-tress* (árboles de sufijos que podemos ver en la Figura 2) donde se representan todos los sufijos de las secuencias en un árbol en el que comparten sus prefijos comunes. El proceso de construcción del árbol tiene un alto coste en tiempo de proceso, por ello, la variante *slide suffix-trees* permite buscar los MUMs de varias secuencias, construyendo el árbol para una sola secuencia y recorriéndolo después para comparar con el resto.

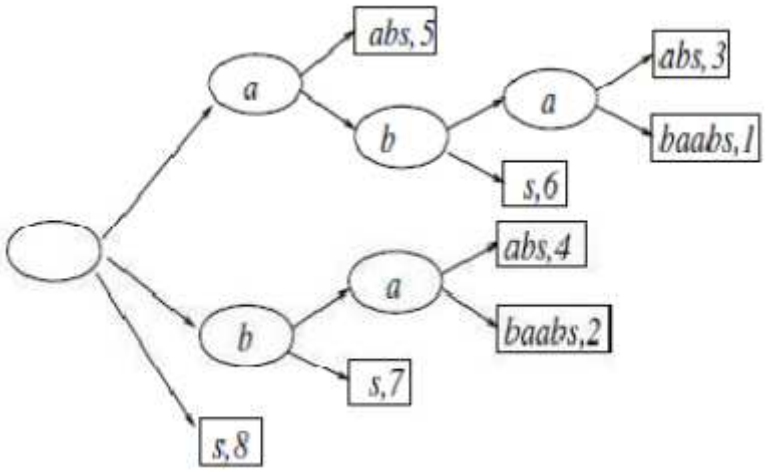


Figura 4 Ejemplo de Suffix Tree

En el proceso evolutivo de las especies, en ocasiones la cadena de un gen muta invirtiéndose completamente ATGCTAGA. => AGATCGTA y aquí aparece el concepto de MUMs inversos, que significa que existe una coincidencia con otra cadena, entendiendo una de las secuencias de ha invertido. En la siguiente figura se puede observar las diferencias entre MUMs directos e inversos:

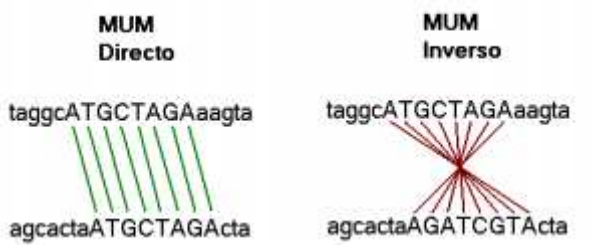


Figura 5 Ejemplo de MUMs Directos e Inversos

[8] Estructura básica de comparación de genomas que tiene significado biológico, que nos permite ver los cambios evolutivos de una especie a otra.

2.6 Matriz de Distancias

Una matriz de distancias es una matriz cuyos elementos representan las distancias entre los puntos, tomados por pares, de un conjunto. Se trata, por lo tanto, de una matriz simétrica de tamaño NxN conteniendo números reales no negativos como elementos. El número N de pares de puntos, (N-1)/2, es el número de elementos independientes en la matriz de distancias.

Supóngase, por ejemplo, el siguiente conjunto de datos a analizar, donde la distancia entre los elementos en píxeles es la métrica que se usara para generar la matriz:

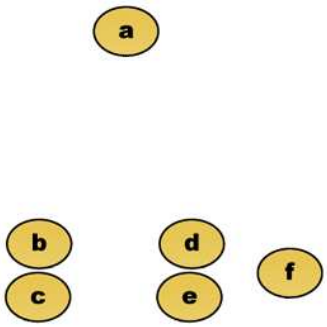


Figura 6 Datos Matriz de Distancias

La matriz de distancias resultante sería:

	a	b	c	d	e	f
a	0	184	222	177	216	231
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	46	83
e	216	128	121	46	0	83
f	231	200	203	83	83	0

En bioinformática se usan las matrices de distancias para representan estructuras de proteínas de una forma independiente de las coordenadas, además de como distancias de emparejamiento entre dos secuencias

2.7 *Minimum Spanning Tree*

Una forma visual de representar información de una matriz de distancias es utilizando un grafo. Pero si lo que interesa es buscar cuales son las relaciones de dicha matriz que tienen un distancia menor hay que buscar una forma de que se unan los nodos con menor peso. Para realizar esto extraeremos un árbol del grafo inicial con la información necesaria. Para realizar esto extraeremos un subárbol del grafo inicial que cumpla las condiciones que requerimos.

El subgrafo final obtenido es el llamado *minimum spanning tree*, un árbol que contiene todos los nodos del grafo inicial pero solo mantendrá las relaciones de entre los nodos menos costosas.

Existen varios algoritmos para saber que nodos son los que se deben seleccionar, uno de ellos es el algoritmo de Prim. Se utilizara el algoritmo de Prim porque es uno de los más sencillos de implementar.

El algoritmo de Prim consiste en ir agregando a cada paso un nuevo nodo al árbol previamente construido. Este nuevo nodo se agregara al árbol con la arista de menor peso.

Para obtener el grafo inicial se puede utilizar los valores de una matriz de distancias. Los nodos se obtendrás de los identificadores almacenadas en la matriz y el peso de los arcos del valor de las relaciones entre dichos identificadores.

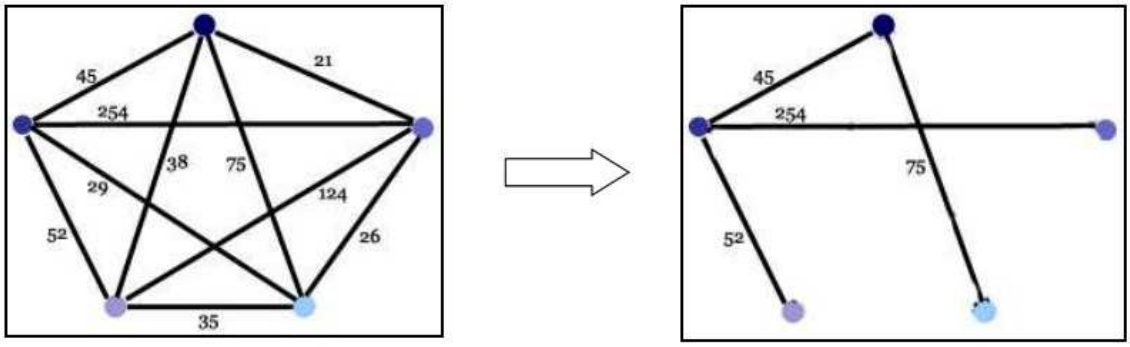


Figura 7 Ejemplo de Grafo Inicial y *minimum spanning tree*

Los pasos del algoritmo Prim son:

1. Se escoge un nodo del grafo inicial y se lo considera el nuevo árbol
2. Se considera la arista con el mínimo peso que une un nodo del árbol con un nodo que no sea del árbol y se une ese nodo al nuevo árbol
3. Si es el número de aristas es igual al número total de nodos menos 1 el algoritmo acaba, sino se vuelve la paso 2.

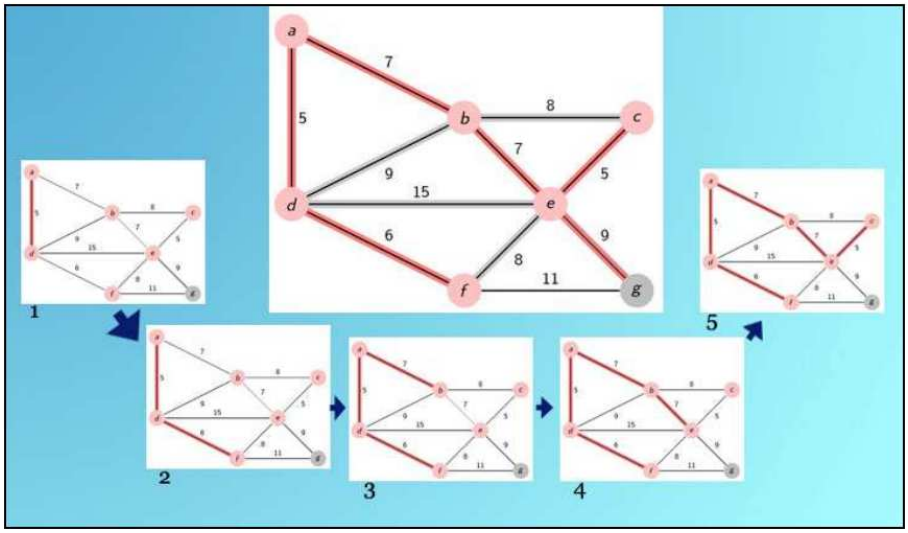


Figura 8 Algoritmo de Prim paso a paso

3. Fases

Para la realización del proyecto organice un sistema de trabajo con diversas fases. Cada una de ellas debe ser completada antes de poder pasar a la siguiente ya que los resultados obtenidos en una fase son necesarios para el inicio de la siguiente. Es un buen sistema a la hora de marcase hitos durante el desarrollo del conjunto del proyecto, de manera que es más fácil el control del tiempo y esfuerzo a dedicar en cada una de las diferentes fases.

Además, teniendo en cuenta la cantidad de información con la que se debe tratar, es muy importante que los juegos de pruebas para la verificación de cada fase sean muy completos. Ya que si debemos repetir los cálculos porque en alguna fase posterior se descubre algún error de alguna de las primeras fases, sería muy costoso tanto en tiempo como en esfuerzo de reparar.

El proyecto se puede dividir en dos partes claramente diferenciadas, en la primera se diseña un nuevo sistema para mejorar la comparación entre los genomas de diferentes especies. En la segunda parte se utiliza dicho sistema para buscar regiones de los genomas que tengan similitudes y así encontrar que especies tienen ancestros comunes.

La diferencia entre las dos partes es aún más evidente si se tiene en cuenta de que como las herramientas del programa están diseñadas para ejecutarse en un servidor, la parte de cálculo de superMUMs se considera pre-proceso, se calcularan los superMUMs de cada comparación una sola vez. Mientras que la segunda parte, la de búsqueda de ancestros comunes está diseñada para que la utilice las aplicaciones on-line, para recalcularse una y otra vez.

A continuación se detallaran las diferentes fases por las que se ha pasado durante la realización del proyecto de búsqueda de ancestros comunes.

3.1 Calculo de superMUMs

En esta fase se tratara el diseño y la construcción del algoritmo encargado de generar las súper-estructuras de MUMs conocidas como superMUMs. También se expondrán los problemas que tiene el algoritmo de cálculo original y como se han solucionado. En el siguiente paso se detalla la creación de una herramienta para la automatización del cálculo de los superMUMs y finalmente se listaran los diversos casos de uso a los que ha sido sometido el algoritmo para su correcta verificación.

Para generar los superMUMs de una comparación entre genomas se utilizaran los datos de los MUMs calculados previamente para la misma comparación. Es decir que se utilizara un algoritmo cuyos parámetros de entrada serán los ficheros donde están almacenados los MUMs.

La información de los MUMs que se almacena en los ficheros es la posición inicial en el genoma de la primera especie, la posición del genoma en la segunda y el tamaño[9] que tiene el MUM.

Para crear un superMUM hay que buscar y agrupar los MUMs cercanos siempre que cumplan la siguiente condición, la suma del tamaño de dos MUMs que se quieran unir debe ser mayor a la distancia[10] que les separa.

(Tamaño MUM1 + Tamaño MUM2) > (Posición Final MUM1 – Posición Inicial MUM2)
Formula1 Calculo de superMUMs

Una vez está clara la fórmula de superMUMs, queda crear un algoritmo que la aplique a todos los MUMs de un fichero. Los pasos que tiene que realizar el algoritmo son los siguientes:

1. Leer los MUMs del archivo de entrada y almacenarlos en una lista en memoria.
2. Recorrer la lista de MUMs, el MUM actual será llamado *current*. Para cada *current* se intentara abrir un superMUM nuevo.
3. El proceso para abrir un nuevo superMUM:
   - 3.1 Recorrer la lista de MUMs de forma independiente al programa inicial aunque solo desde la posición del *current* hacia adelante.
   - 3.2 Ir aplicando la fórmula de superMUMs entre el ultimo MUM añadido al superMUM y los siguientes MUMs de la lista
   - 3.3 Cuando no quede ninguno que cumpla la condición. Entonces ese superMUM se cerrará.
   - 3.4 Antes de guardar el nuevo superMUM se comprobara que no sea un superMUM absorbido comparándolo con los superMUMs ya calculados. Si el nuevo superMUM está contenido completamente en alguno de los superMUMs existentes, será descartado.[11]
4. Los MUMs “usados” para crear un superMUM son marcados como MUMs absorbidos.
5. Al terminar de recorrer la lista, se volcará la lista de superMUMs en un fichero de salida. En ese mismo fichero también se añadirán los MUMs que no estén marcados como absorbidos, es decir los MUMs no absorbidos.

[9] Consideramos como tamaño el número bases que forma el MUM.
[10] Tomamos como distancia la diferencia entre la posición final del primer MUM y la inicial del segundo
[11] El problema de los superMUMs absorbidos esta detallado en la sección de estrategias de superMUMs

6. Finalmente, también se generara un fichero con las estadísticas del cálculo, número de superMUMs creados, suma del tamaño de todos los superMUMs y número de MUMs no absorbidos.

Es importante tener en cuenta que del cálculo de MUMs de una comparación entre genomas se obtienen dos archivos resultantes, uno donde se almacenan los MUMs directos y en el otro los MUMs inversos[12]. Por tanto, cuando se haga el cálculo de superMUMs se consideraran los dos archivos de MUMs por separado.

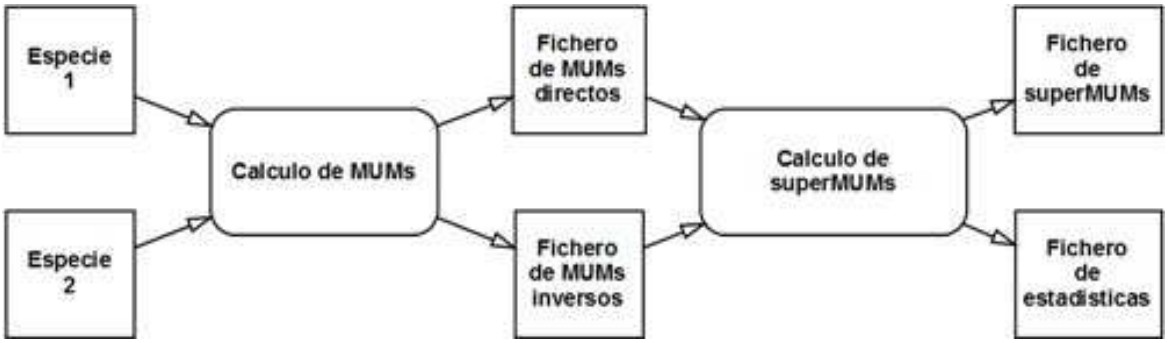


Figura 9 Esquema de la comparación entre especies

3.1.1 SuperMUMs: Casos de Estudio

Debido a la gran cantidad de MUMs generados incluso en las especies con el genoma más pequeño, comprobar si los cálculos de superMUMs son correctos para especies reales es difícil de demostrar.

Por tanto, crearon casos de ejemplo de poco tamaño para que fuera sencilla su comprobación. Casos donde mostraran todas las posibles situaciones para el cálculo de superMUMs:

- Creación de superMUMs estándar

En este caso podemos observar la agrupación de 5 MUMs en dos superMUMs diferentes, la razón por la que no se unan los 5 en un único superMUM es que la distancia entre el tercero y cuarto MUM es demasiado grande para que se cumpla la formula.

[12] Ver la sección de MUMs en fundamentos teóricos.

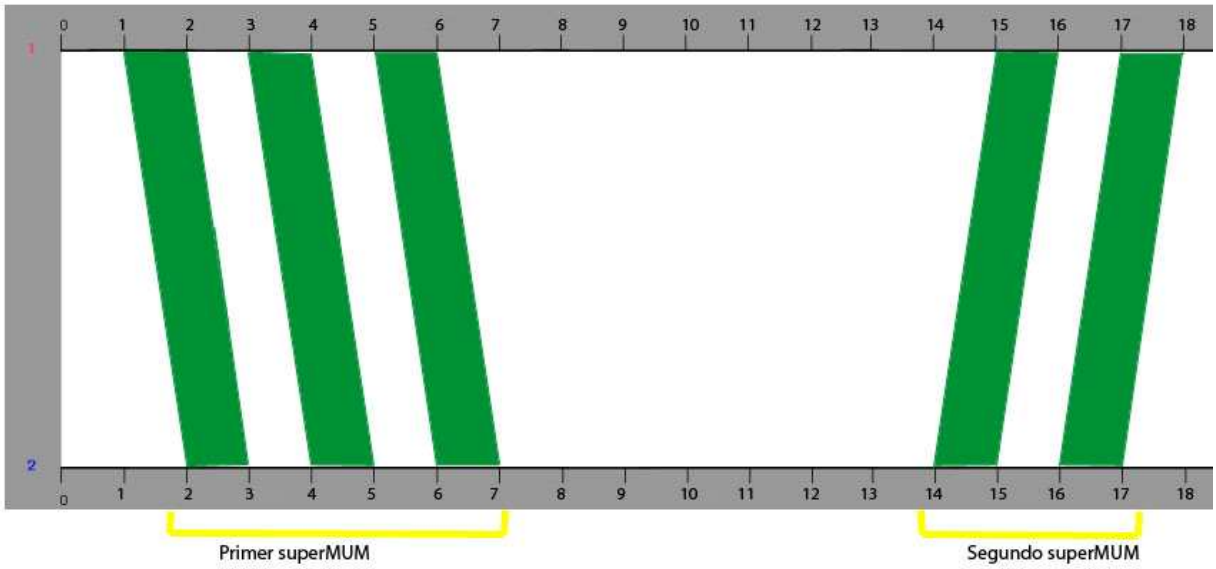


Figura 10 Ejemplo de superMUMs sencillos

- SuperMUMs con MUMs no absorbidos

En este caso comprobamos que hay MUMs que aunque tengan una parte de su genoma dentro de un superMUM, si la segunda parte queda fuera, se consideran MUMs no absorbidos. Se consideran MUMs no absorbidos como aquellos MUMs que no forman parte de ningún superMUM. Estos MUMs serán volcados en el fichero de salida junto con los superMUMs finales. En el ejemplo vemos que los MUMs tercero y quinto de este sistema no son absorbidos.

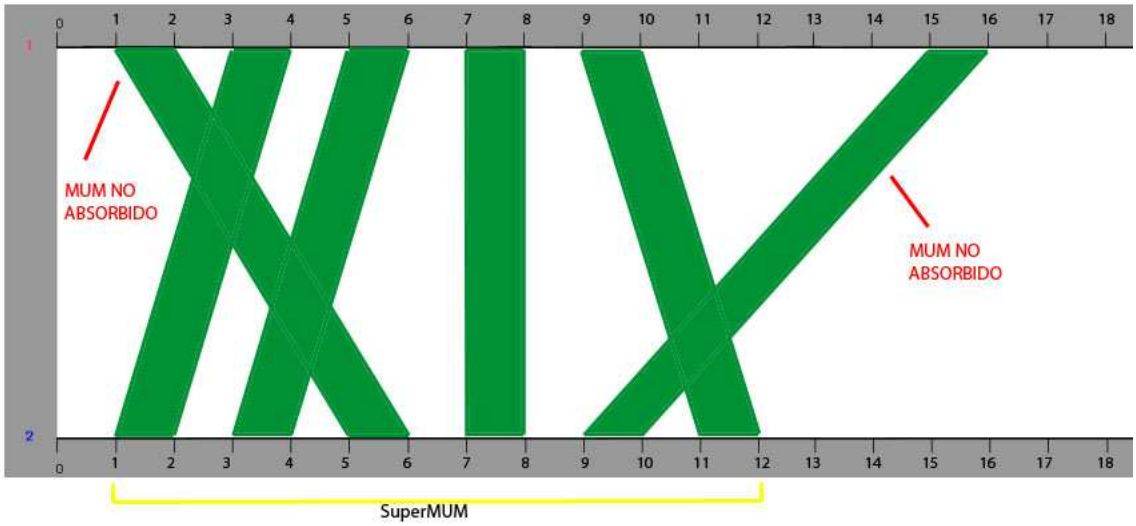


Figura 11 Ejemplo de MUMs no absorbidos

- SuperMUMs con partes en común

Es como el caso anterior, pero algo más extremo. Los MUMs no absorbidos que en un genoma están dentro de otro superMUM, forman un superMUM independiente.

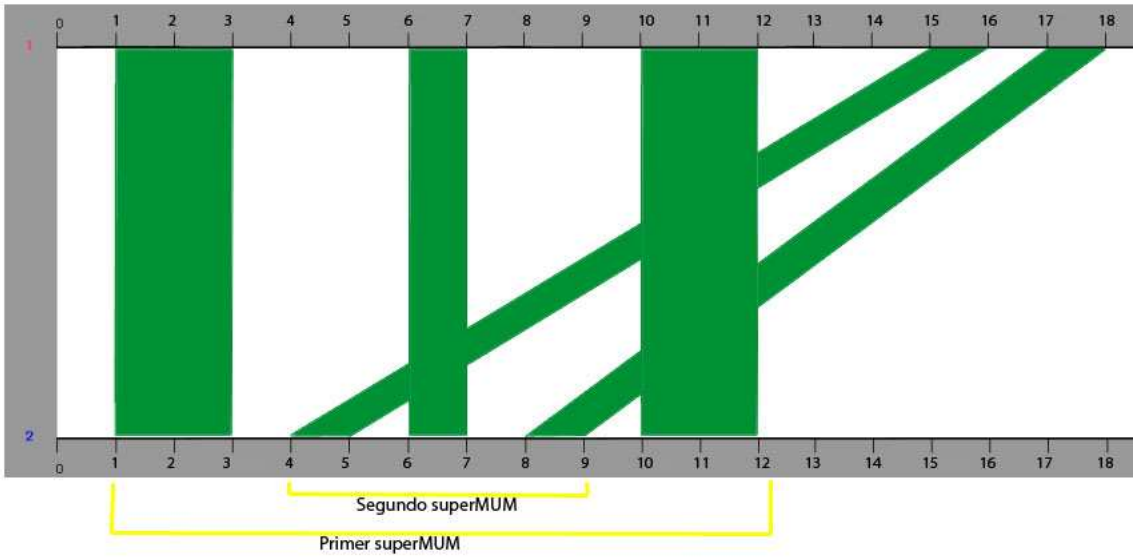


Figura 12 Ejemplo de superMUMs solapados

- SuperMUMs inversos

Como su nombre indica, los MUMs inversos están formados por la misma secuencia de bases de un genoma invertida en el otro genoma. Pero para el cálculo de superMUMs esto no nos afecta, se les aplica el mismo algoritmo que para los superMUMs directos. Se los suele representar en rojo.

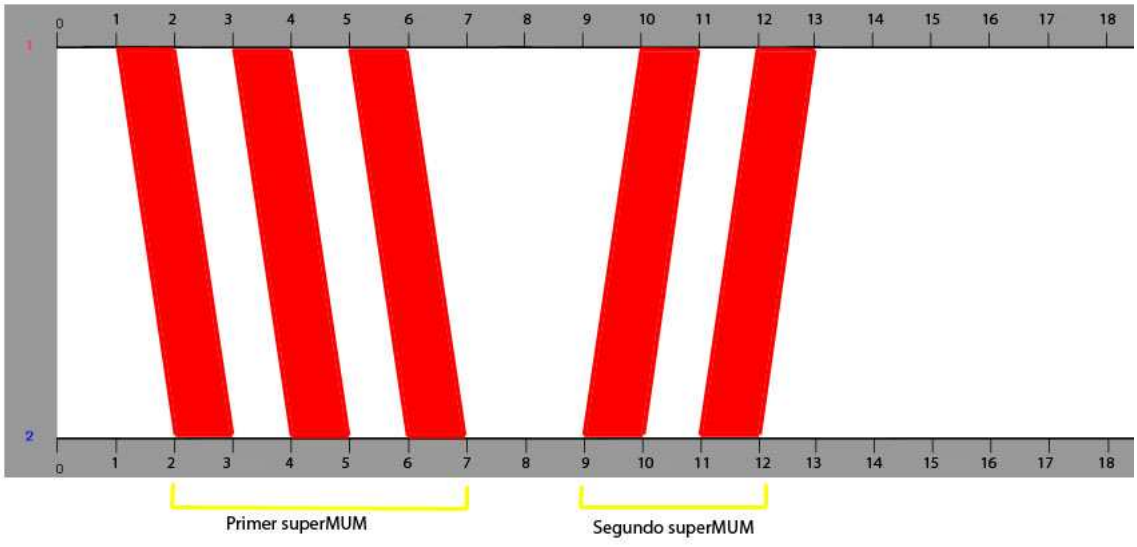


Figura 13 Ejemplo de superMUMs inversos

Cuando el resultado con los casos de prueba fue satisfactorio y se comprobó que los superMUMs resultantes estaban correctamente formados, se aplicaron los cálculos a ficheros de MUMs reales para medir el tiempo de cómputo.

3.1.2 Estrategias Seguidas para el Desarrollo del Cálculo de SuperMUMs

- Espacios Entre MUMs

Debido al tamaño de algunos genomas la distancia entre MUMs a veces es excesiva para poder crear superMUMs, de manera que la fórmula del algoritmo original no es suficiente para especies del dominio Eucariota.

Solución propuesta:

- Añadir un multiplicador para la fórmula de superMUMs

Modificando la fórmula para que acepte un nuevo parámetro, un multiplicador de tamaño. Ahora para generar nuevos superMUMs se comprobará que la suma del tamaño de los MUMs multiplicada por el nuevo parámetro debe ser mayor que la distancia entre los MUMs comparados

Con este nuevo parámetro podemos ayudar a la generación de superMUMs en especies con tamaños de genomas enormes. Se ha de tener cuidado en utilizar un multiplicador demasiado grande, ya que si es así no se crean nuevos superMUMs sino que se unen los ya creados. Después de los juegos de pruebas se llegó a la conclusión de que el multiplicador correcto para las Archaea era de 2.5

(Tamaño MUM1 + Tamaño MUM2)*Multiplicador > (Posición Final MUM1 – Posición Inicial MUM2)

Formula1.1 Calculo de superMUMs

- SuperMUMs absorbidos o solapados

Existe la posibilidad de que algunos superMUMs estén solapados entre sí. Si solo es parcialmente se consideraran superMUMs distintos, pero si un superMUM está contenido dentro de otro no deberá guardarse en la lista de superMUMs.

Soluciones propuestas:

- Descartar superMUMs solapados completamente (absorbidos)

La solución consiste en hacer comprobación justo antes de añadir un superMUM recién formado en la lista de superMUMs finales. Se comprueba si los extremos del nuevo superMUM están dentro de alguno de los superMUMs ya listados. Si es así, el nuevo superMUM se descarta directamente.

La comprobación de los superMUMs absorbidos es muy costosa, teniendo en cuenta que se debe realizar para todos los superMUMs finales. Para hacerla algo más eficiente, se utilizara el sistema de dos listas enlazadas[13] para que solo deba compararse alguno de los superMUMs.

[13] Este sistema de dos listas esta detallado en una de las soluciones para la reducción de tiempo de cómputo.

- Unir superMUMs solapados

A parte de la absorción completa de un superMUM por otro se debe tener en cuenta cuando un superMUM solo tiene una parta de su longitud de entro de otro superMUM. Normalmente se deberán considerar como dos superMUMs independientes, pero se consideraran como uno solo si tienen más del 20% de su longitud en común.

A este proceso le llamaremos fusión de superMUMs, se trata de unir como uno solo dos superMUMs parcialmente solapados. Para la creación del nuevo superMUM se tendrán en cuenta la suma de las longitudes de ambos superMUMs solapados pero con cuidado de contar la región solapada una sola vez.

- Tiempo de cómputo

Debido a la gran cantidad de MUMs que pueden generarse en algunas comparaciones, sobre todo al trabajar con Eucariotas que pueden rondar en torno a los 80 millones de MUMs, el algoritmo tiene mejorar la eficiencia para reducir el tiempo de cálculo.

Soluciones propuestas:

- Limitar la búsqueda de MUMs para unir a superMUMs

Cuando se buscaban MUMs para crear superMUMs se recorría la lista desde el *current* hacia el final. Esto implicaba un coste de tiempo muy grande, sobre todo cuando *current* estaba cerca de las primeras posiciones.

Para solucionarlo se almacena el tamaño del mayor MUM del fichero de entrada, cuando se recorre la lista de MUMs en busca de candidatos para el superMUM se aplica una condición, si la distancia entre el final del superMUM y el siguiente MUM candidato es mayor a la suma entre el tamaño del MUM final del superMUM y el del mayor MUM se cancela la búsqueda. Ya que damos por sentado de que si la fórmula de superMUMs no se cumple con el MUM más grande tampoco lo hará con ningún otro candidato y por tanto no tiene sentido seguir buscando.

- Uso de listas enlazadas para almacenar los superMUMs cerrados

El tiempo excesivo al calcular superMUMs puede ser causado también por una mala gestión de la memoria donde se almacenan estos. Por eso, el uso de una *array* de estructuras para guardar los superMUMs finales no es eficiente.

Para solucionarlo modificamos esa *array* por una lista enlazada, donde cada nodo tiene la información de un superMUM y un puntero al siguiente elemento. De esta manera la memoria ocupada por los superMUMs no tiene que ser consecutiva[14] y evitamos posibles errores por falta de esta. Esta nueva lista será llamada lista de superMUMs finales.

[14] La memoria que ocupan los elementos de un *array* es siempre consecutiva, en bloque.

Finalmente se añadió una segunda lista que nos ayude a acceder a elementos concretos la lista de superMUMs finales sin tener que recorrerla desde el principio[15]. Los nodos de esta lista no contienen información, solo apuntan a los elementos concretos de la primera. Esta segunda lista será llamada la lista de accesos.

El sistema de las dos listas funciona de la siguiente manera,

1. Al generase un nuevo superMUMs se añade a la lista de superMUMs finales. También se añade un nuevo nodo a la lista de accesos con un apuntador hacia el nuevo superMUM.

2. En cada movimiento del *current* se intentara eliminar superMUMs de la lista de accesos. En el momento en que las posiciones iniciales del *current* sean mayores que las posiciones finales de alguno de los superMUMs que estén en la lista de accesos, ese superMUM será eliminado solo de la lista de accesos, permaneciendo inalterado en la lista de superMUMs

La lista de accesos nos permitirá un acceso casi instantáneo a los superMUMs recién creado, de esta manera las comparaciones que necesiten leer sus valores, como el descarte de superMUMs absorbidos, **no deberán recorrer toda la lista de superMUMs**. Esto reduce mucho el tiempo de cómputo de todo el algoritmo.

En la figura 14 vemos las dos listas, en la de superMUMs finales hay cinco superMUMs almacenados, en la lista de acceso se apunta al segundo y cuarto superMUM de la primera lista.

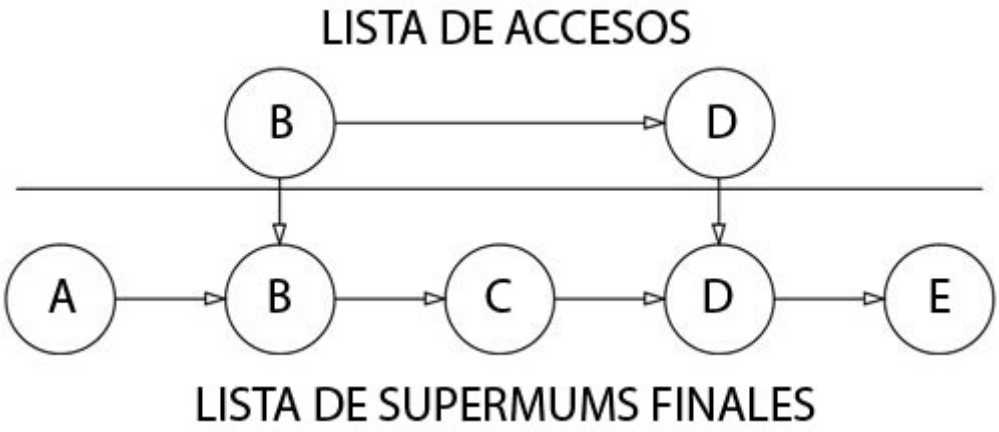


Figura 14 Ejemplo de dos listas. Se muestran los superMUMs A,B,C,D y E

[15] Las listas enlazadas solo tienen acceso secuencial, para buscar un elemento concreto hay que recorrerla de inicio a fin.

- Evitar abrir nuevos superMUMs que posiblemente estarán absorbidos

Aunque ya se ha añadido una condición que impide que se almacenen superMUMs absorbidos[16], se sigue perdiendo tiempo generándolos para luego descartarlos. Hay que buscar una manera para poder evitar que se intenten abrir.

La solución adoptada es comparar la inclinación del MUM *current* con los MUMs del inicio y final de todos los superMUMs posiblemente absorbentes[17]. Si los MUMs comparados son paralelos, el superMUM resultante será absorbido y por tanto no se inicia el proceso de abrirlo. Si el ángulo de la comparación es más abierto o cerrado sigue adelante con el proceso.

3.1.3 Automatización del Cálculo de SuperMUMs

Ahora que el algoritmo ya ha sido implementado se ha de automatizar para que se calculen los superMUMs de todos los ficheros de MUMs dentro de una misma carpeta. Para ello se ha creado una lanzadera de superMUMs, un programa que solo necesita el nombre de la carpeta con los MUMs y el valor del multiplicador que se quiera aplicar para que calcule los superMUMs de todos los ficheros que haya en esa carpeta sean directos o inversos.

Originalmente se diseñó para que los superMUMs de diferentes archivos fueran calculados en paralelo para así ahorrar tiempo. Pero debido al extremo uso de memoria que se necesita para el cálculo de superMUMs, sobretodo con las Eucariotas, se ha preferido hacer un cálculo lineal, fichero tras fichero. Esto convierte el proceso más lento pero evita los errores por falta de memoria.

3.2 Obtención de Ancestros Comunes

Una vez obtenidas las comparaciones entre especies con los superMUMs es momento de buscar similitudes entre las regiones de los genomas de estas especies. Esa es la base de la búsqueda de ancestros en común. Primero hay que buscar el grado de conservación entre la región especificada del genoma de la especie deseada y todas las demás. Se generara una matriz de similitudes comparando todas con todas las regiones conservadas en el resto de genomas del resultado anterior. Finalmente se utilizara la matriz resultante para generar un árbol del tipo *mínimum spanning tree* que mostrara, para la región especificada del genoma de la especie deseada, el grado de conservación con el resto de especies.

[16] Solapados completamente en otro superMUM
[17] Los superMUMs que aun tengan punteros en la segunda lista enlazada

3.2.1 Búsqueda de una Región Conservada en un Segundo Genoma

Necesitamos crear una herramienta que permita calcular la conservación de los superMUMs de la región del genoma de una especie en esa misma región del genoma de otra especie.

Para que el programa funcione correctamente se le debe especificar los dos genomas con los que queremos trabajar (que llamaremos primer y segundo genoma respectivamente) y la región de la cual queremos obtener el grado de conservación. Aunque solo la región del primer genoma será introducida por el usuario.

Para el cálculo del grado de conservación utilizaremos una formula sencilla, la suma de las longitudes de los superMUMs conservados de la región dividido por la longitud de la región. Se sumara el resultado de ambas operaciones (una por cada genoma) y ese será el grado de conservación.

$$\frac{\textbf{Suma longitudes sMUMs conservados1}}{\textbf{Longitud Region1}} + \frac{\textbf{Suma longitudes sMUMs conservados2}}{\textbf{Longitud Region2}}$$

Fórmula2 Calculo del grado de conservación

El algoritmo diseñado para que realice estas funciones será el siguiente:

1. Buscar todos los superMUMs que pertenezcan a la región del primer genoma, estos superMUMs serán los conservados.
2. Calcular la región del segundo genoma a partir de los superMUMs conservados. El superMUM con la posición más baja en el segundo genoma marcara el inicio de la región y el superMUM con la posición más alta será el final.
3. Aplicamos la fórmula del cálculo de grado de conservación, almacenaremos el resultado y la región del segundo genoma.
4. Buscaremos las agrupaciones de superMUMs en los extremos de las regiones, eliminaremos los superMUMs de la agrupación con el menor tamaño.
5. Si ya no quedan más superMUMs después del paso anterior se sale del programa, si no volvemos al paso 2.

En la figura 14 vemos que la inclinación de los superMUMs en el genoma 2 perjudica al grado de conservación por tener una región tan grande en comparación a la longitud de los superMUMs. Es un buen ejemplo para ver cuando es conveniente eliminar una agrupación de superMUMs conservados.

La región del primer genoma no varía tras la eliminación, por eso nos sirve para saber si es conveniente realizar la eliminación para obtener mejores grados o no.

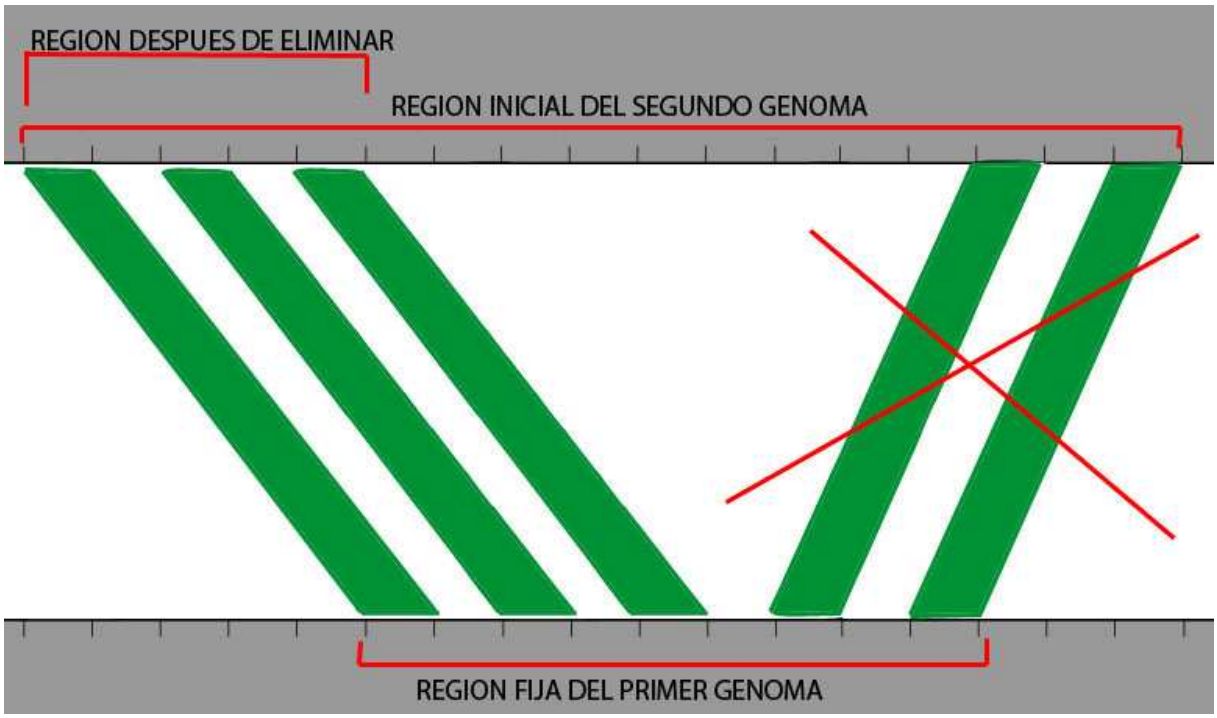


Figura 15 Ejemplo de cuando es conveniente eliminar superMUMs

En la figura 15 vemos un ejemplo de cuando no es conveniente eliminar superMUMs conservados. El cálculo de grado de superMUMs conservados original será mayor que el grado obtenido después de la eliminación.

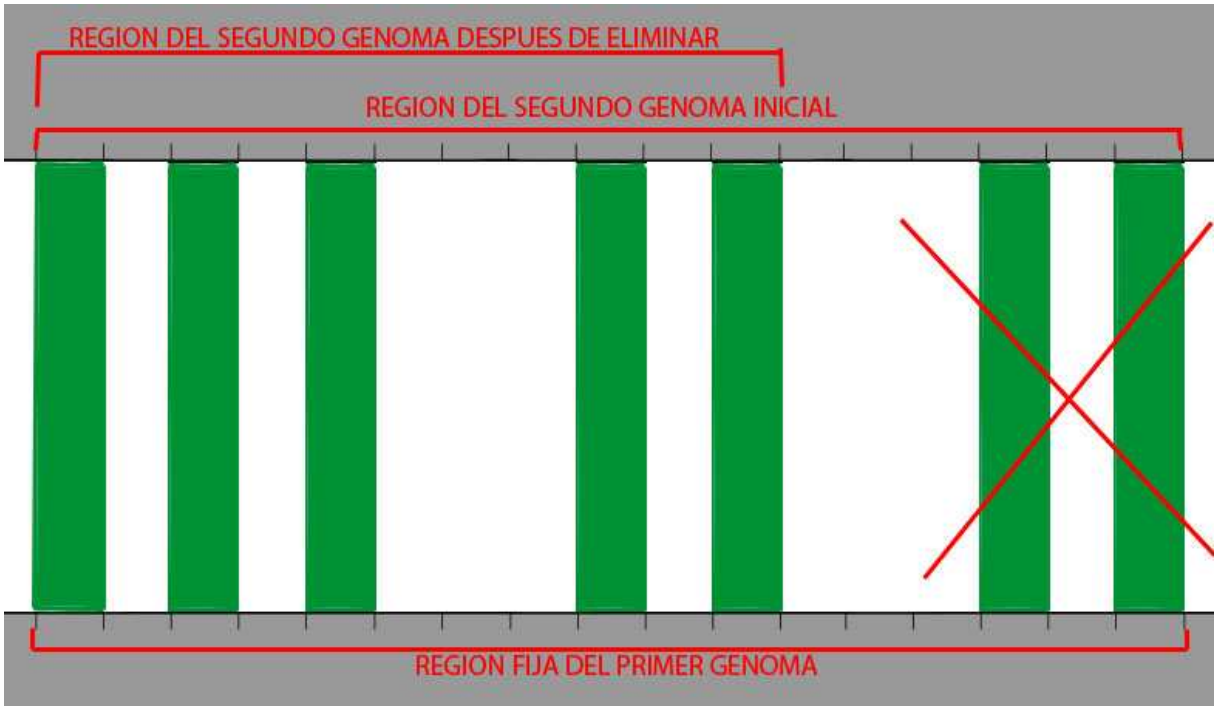


Figura 16 Ejemplo de mala eliminación de agrupaciones de superMUMs

3.2.1.1 Regiones Conservadas: Casos de Estudio

Para los juegos de prueba se han utilizado los mismos casos que para el cálculo de superMUMs, ya que para las agrupaciones de superMUMs conservadas se usa el mismo sistema que para unir MUMs en un superMUM.

3.2.1.2 Estrategias Seguidas para el Cálculo de Regiones Conservadas

- Posibles superMUMs conservados cortados por el extremo de una región

Se debe tener en cuenta también los superMUMs que quedan cortados por los extremos de la región inicial, en ese caso no se pueden considerar como conservados porque hay una parte de ellos que no entra en la región, pero tampoco se pueden descartar porque estaríamos perdiendo información útil.

Solución Propuesta:

Para evitar problemas se trataría de considerar superMUM conservado unicamente la zona de este que quede dentro de la región inicial. Es decir que se creara un nuevo superMUM a partir del tamaño útil del anterior.

- Solapamiento de superMUMs conservados

Debido a que buscando superMUMs conservados se deben tener en cuenta tanto los superMUMs directos e inversos de la comparación entre los genomas es posible que al buscar los superMUMs conservados dentro de una región haya algunos que se solapen entre sí. Esto provocaría que el resultado de la fórmula de grado de conservación fuera erróneo, ya que las longitudes solapadas deben constar solo una vez.

Solución Propuesta:

Para evitar los solapamientos se ha creado dos listas binarias[18] con la misma longitud que las regiones. A medida que se vayan leyendo los superMUMs conservados se irán marcando las respectivas posiciones en la lista binaria. De esta manera, aunque haya diferentes superMUMs compartiendo las mismas posiciones, solo sumaran como usados una vez.

3.2.1.3 Búsqueda de una Región Conservada en el Resto de Genomas

Para encontrar los ancestros comunes correctamente, se ha de buscar las regiones conservadas de una especie en concreto en relación a todas las demás especies del mismo dominio. Para ello se ha creado un programa que, para una especie y región especificada, aplique la búsqueda de regiones conservadas a todas las demás especies.

Los grados de conservación y las regiones obtenidas son almacenados en un fichero de salida que tendrá de nombre el identificador de la especie[19] a la que corresponde el genoma y

[18] Utilizaremos el 0 binario para marcar como espacio vacío, y el 1 para marcar el usado.
[19] Los nombres de las especies son muy largos, se utilizan números enteros para identificarlos y ahorrar espacio

la región especificada por el usuario Añadir el identificador del genoma es un buen mecanismo para poder diferenciar los diferentes ficheros entre si y nombrarlos con las regiones sirve para que se puedan hacer diferentes consultas simultaneas del mismo fichero con otras regiones sin que se sobrescriban dichos ficheros.

3.2.2 Cálculo de Similitud entre las Regiones Conservadas de dos Genomas

Se han calculado las regiones conservadas para todas las especies respecto a la especie elegida por el usuario. Ahora se deben comparar esas regiones conservadas entre sí para buscar cuales tienen más similitudes.

Para poder realizar el cálculo utilizaremos un programa muy similar al del paso anterior. Pero con la diferencia de que las dos regiones a comparar son leídas del fichero de entrada y ninguna introducida por el usuario.

El programa debe leer el fichero de entrada, almacenar en memoria todos los identificadores de genomas que haya y sus respectivas regiones. Después ira cogiendo los genomas por parejas, abrirá los archivos de comparaciones de superMUMs de estos genomas y buscara los superMUMs que estén dentro de ambas regiones a la vez. Estos superMUMs serán los superMUMs conservados para esas especies.

Cuando se hayan encontrado los superMUMs conservados, se calculara el grado de conservación entre ambos genomas de forma similar a la fase anterior pero no hace falta reducir el número de superMUMs ya que la longitud de la región del segundo genoma es constante. Aplicando la misma fórmula para el cálculo del grado y utilizando también el sistema con los dos *arrays* binarios para evitar solapamiento entre superMUMs directos e inversos:

$$\frac{\text{Suma longitudes sMUMs conservados1}}{\text{Longitud Region1}} + \frac{\text{Suma longitudes sMUMs conservados2}}{\text{Longitud Region2}}$$

Fórmula3 Calculo del grado de conservación

El valor resultante de la aplicación de la formula se almacenara en un fichero de salida junto con los identificadores de los dos genomas comparados.

3.2.2.1 Cálculo de Similitud entre Regiones: Casos de Estudio

Para los juegos de prueba se han utilizado los mismos casos que para el cálculo de regiones conservadas. Aprovechando los cálculos de regiones y grados para encontrar sus respectivas matrices de distancias.

3.2.2.2 Estrategias Seguidas para Cálculo de Similitud entre las Regiones

- SuperMUMs conservados cortados por los extremos de las regiones

Igual que sucede para el cálculo de regiones conservadas se ha de tener en cuenta que las regiones de ambos genomas pueden cortar alguno de los posibles superMUMs conservados. Si eso ocurre solo se debe tener en cuenta la parte del superMUM incluida dentro de la región. Pero en esta fase es algo más complicado que en la anterior ya que antes de considerar un superMUM conservado se ha de comprobar que no sea cortado por ninguna de las dos regiones.

Posible Solución:

Se añadirá una comparación extra al código. Antes de considerar un superMUM como conservado deberá comprobarse que ninguna parte de su longitud es cortada por ninguno de los extremos de las dos regiones. (Serán 4 extremos a comparar en total). Si se cumple esto, el superMUM será conservado sin más. Pero si algún extremo lo corta, se tendrá que calcular un nuevo superMUM conservado que tenga como longitud únicamente la parte que este dentro de ambas regiones.

3.2.3 Cálculo de la Matriz de Similitudes[20] entre las Regiones Conservadas

Este paso se basa simplemente en la automatización del cálculo de Similitud entre las regiones conservadas. Este programa deberá leer el fichero obtenido de la búsqueda de conservación de regiones para encontrar todas las especies que deberá comparar. Cuando sepa que especies son, aplicara el cálculo de similitud entre las regiones conservadas para todas ellas.

El resultado que se obtendrá con este programa será una lista con todas las especies que conservan una determinada región y el valor de similitud que hay entre ellas. Con esta lista se puede construir la matriz de similitudes.

Un ejemplo de matriz para 3 genomas (A, B y C) seria:

	A	B	C
A	–	GradoAB	GradoAC
B	–	–	GradoBC
C	–	–	–

Figura 17 Ejemplo de Matriz de Similitudes para 3 genomas

[20] Matriz del tipo distancias (Ver Sección Fundamentos Teóricos) pero los valores que relacionaran los genomas entre si serán sus grados de conservación.

Es interesante observar que no importa el orden cuando comparamos los genomas, el grado de similitud entre sus regiones será el mismo tanto si hacemos la comparación entre A y B como entre B y A.

3.2.4 Construcción del *Minimum Spanning Tree* con la Matriz de Similitudes

Ahora que las regiones de diferentes genomas están relacionadas entre sí hay que representarlos gráficamente. Para ello utilizaremos el fichero generado de matriz de similitudes para construir un grafo de manera que se representen los diferentes genomas como nodos y las correlaciones como aristas. Luego se aplicara el algoritmo de Prim[21]para obtener el *minimum spanning tree*.

Figura 18 Uso del Algoritmo de Prim

De esta manera tendremos una representación de cómo se conserva la región de un genoma en el resto de los genomas de todas las especies de su dominio.

3.3 Planificación Temporal

A continuación se verán las diferencias en la planificación que se hizo en el momento de entregar el trabajo previo del proyecto y el tiempo real que se ha necesitado para cumplir los objetivos.

Se puede apreciar que la principal diferencia está en el tiempo invertido para optimizar el algoritmo de cálculo de superMUMs, para adaptarlos al trabajo con Eucariotas, durante los meses de junio y julio.

	Planificación Previa	Tiempo Real
Diciembre 2009, Enero, Febrero 2010	Diseñar el programa del cálculo de superMUMs	Diseñar algoritmo para cálculo de superMUMs e implementarlo
Marzo 2010	Diseñar el programa de búsqueda de conservación de una región en otro genoma	Crear juegos de pruebas para el cálculo de superMUMs y arreglar errores.
Abril 2010	Programa de cálculo de similitudes todos con todos	Diseñar el algoritmo de búsqueda de conservación de región en otro genoma e implementarlo

[21] Esta detallado en la sección de Fundamentos Teóricos

Mayo 2010		Programa de comparación de similitud de regiones todos con todos
Junio 2010	Búsqueda del *minimum Spanning Tree* a partir de la matriz de similitudes	Búsqueda del *minimum Spanning Tree* a partir de la matriz de similitudes
Julio 2010		Optimización para los superMUMs de Eucariotas
Agosto 2010	Documentación	Documentación
Septiembre 2010	Presentación del Proyecto	Presentación del Proyecto

4. Resultados y Discusión

Se han hecho los cálculos para los tres tipos de dominio de especies, Archaea, Bacteria y Eucariota. Como lo más importante para esta parte son el número de superMUMs obtenidos y el tiempo de cálculo utilizado se van a mostrar esos datos en sendas gráficas. De esta manera más fácil comparar los resultados obtenidos para diferentes especies.

El primer grafico es una comparación entre la suma de superMUMs, tanto directos como inversos, de los diferentes dominios de especies, menos de Eucariotas, que representa el número de superMUMs de una sola comparación. (La de humano y Macaco).

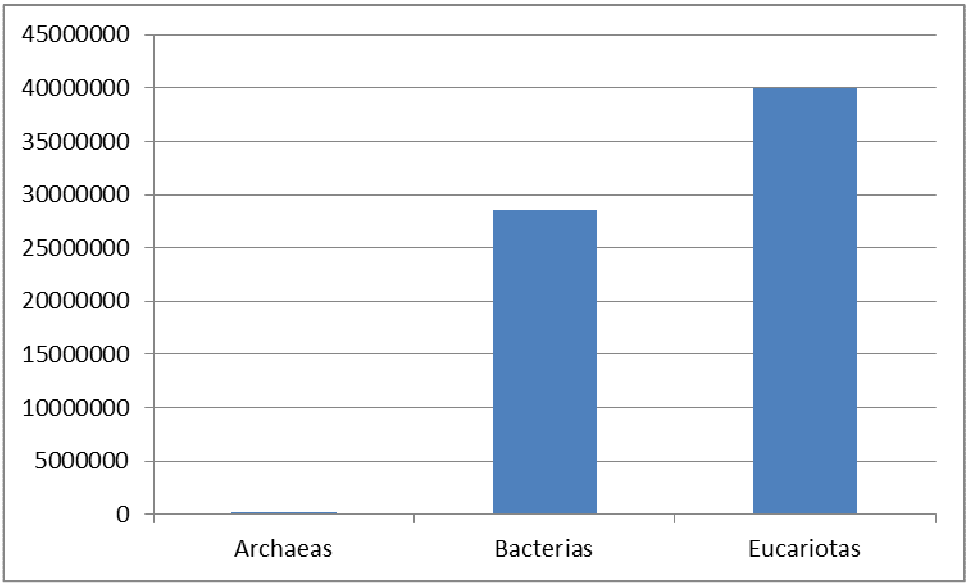


Figura 19 Grafico donde se muestran la cantidad de superMUMs por cada dominio de especies

Aunque en comparación con las demás especies las Archaeas tengan un número insignificante de superMUMs, tienen aproximadamente 16000.

Es interesante observar la cantidad de información que obtenemos del estudio de las Eucariotas en comparación con especies menos desarrolladas.

El segundo grafico representa el tiempo de cálculo en segundos de todas las especies de los dominios de Archaea y Bacteria frente al tiempo para el cálculo de solo una especie de Eucariota.

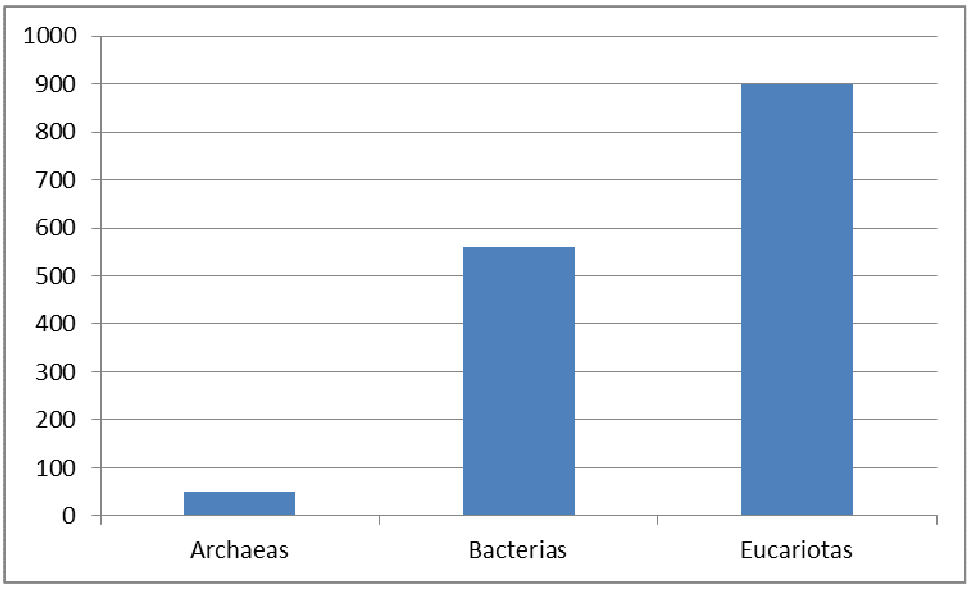


Figura 20 Grafico donde observan el tiempo de cálculo en segundos para la generación de superMUMs

Las optimizaciones realizadas en el cálculo de superMUMs han sido necesarias para que el tiempo de generación de superMUMs para la comparación de las Eucariotas sea viable. En la figura 20 se puede observar la enorme diferencia entre la comparación de las especies humana y macaco. Sin optimizar se tardaban más de 5 días, con la optimización solamente unos 15 minutos.

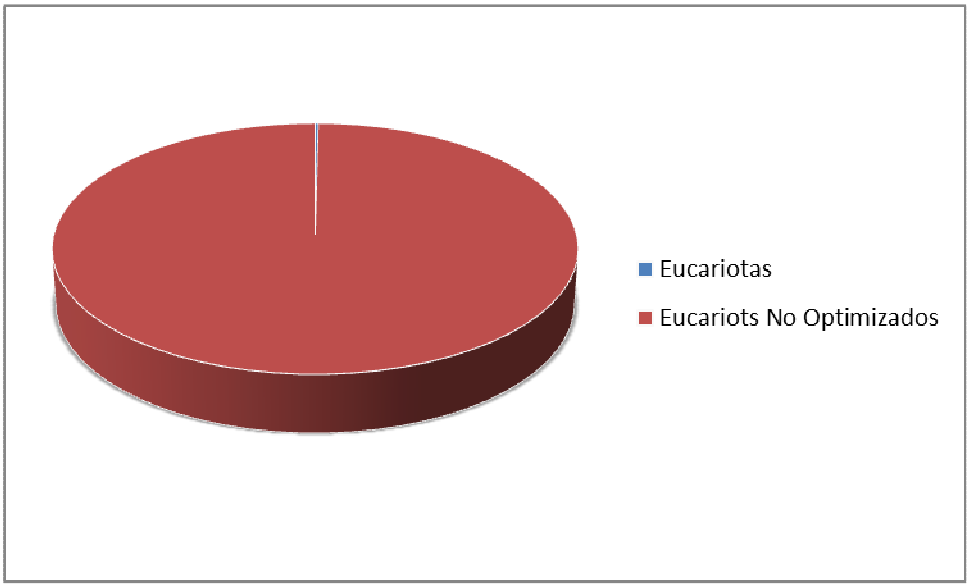


Figura 21 Mejora en la Optimización de Eucariotas

Para encontrar que multiplicador deberíamos considerar como estándar en el cálculo de superMUMs comparé el número de superMUMs y de MUMs no absorbidos obtenido con diferentes multiplicadores. Utilice el 1, 1.5, 2, 2.5, 3, 3.5 y 4.

Valor del Multiplicador	Numero de SuperMUMs	Numero de MUMs no Absorbidos
1	59174	145630
1.5	95527	132433
2	136189	124353
2.5	**160150**	**119025**
3	185817	114905
3.5	209625	111749
4	236131	109221

Elegí utilizar el multiplicador de 2.5 porque estudiando los resultados se puede observar que aunque en los primeros casos a medida que aumentamos los el valor del multiplicador se crean muchos nuevos superMUMs y el número de MUMs no absorbidos se va reduciendo en proporción. Pero pasado el multiplicador de 2.5 aunque siguen incrementando el número de superMUMs el número de MUMs absorbidos ya no se reduce tanto, eso induce a pensar que con **multiplicadores altos no se generan nuevos superMUMs solo se unen entre si los ya existentes.**

También se debía demostrar que los superMUMs son unas buenas unidades para la comparación de dos genomas, genere dos matrices de similitud de la misma comparación en una solo se tenía en cuenta el número de superMUMs para cada relación mientras que en la otra matriz se tenía en cuenta tanto el número de superMUMs como el número de MUMs no absorbidos.

Al ordenar los ficheros de las matrices de simulación por el valor de la comparación (De mayor a menor) y comparar los dos ficheros entre si obtuve las siguientes diferencias:

15 16 1046655	15 16 1087401
16 17 992325	16 17 1046329
15 17 892499	15 17 946683
16 18 722690	16 18 791440
15 18 707433	15 18 772832
17 18 667682	17 18 740784
24 26 349856	24 26 534963
24 25 217371	24 25 384433
25 26 109444	25 26 227149
17 19 75000	17 19 142423
18 19 72942	18 19 141956
15 19 67907	15 19 135660
16 19 67050	16 19 133139
38 39 48588	5 30 93150
5 30 32428	38 39 79971
37 39 31763	5 6 74800
33 34 28742	37 39 70315
5 6 21793	6 30 67066
6 30 18571	33 34 65127
37 38 16914	34 35 46042
34 35 14379	42 43 45539
42 43 10917	37 38 45446
7 30 10736	33 35 36534
37 44 10432	13 19 34865
34 36 10229	37 44 29167
33 35 10213	33 36 27389

Figura 22 Comparación entre Dos Matrices, la de la izquierda son solo superMUMs en la derecha son superMUMs y MUMs no absorbidos. Se ha mostrado solo las relaciones con valores más altos

Se puede observar que aunque los valores de superMUMs y MUMs son diferentes, las mejores relaciones entre genomas (Las que están más arriba) se mantienen en las mismas posiciones en ambos ficheros. Esto es muy importante porque significa que **se puede considerar los MUMs no absorbidos del cálculo de superMUMs como ruido.** Así que a partir de ahora en las aplicaciones que se usen superMUMs solo se tendrán en cuenta los superMUMs, y no los MUMs absorbidos.

Finalmente, como ejemplo de la construcción de los árboles de *minimum spanning tree* tenemos una matriz con las relaciones entre ocho genomas de Archaea.

En la figura 23 podemos observar la matriz de similitudes después de la comparación entre las regiones conservadas de los genomas.

	2	3	4	5	6	7	8
2	-	-	-	-	-	-	-
3	50	-	-	-	-	-	-
4	345	823	-	-	-	-	-
5	512	316	674	-	-	-	-
6	294	32	329	142	-	-	-
7	67	4	732	525	1542	-	-
8	354	926	242	249	323	90	-

Figura 23 La Matriz de Similitudes Para las regiones conservadas

Si aplicamos el algoritmo de Prim en la matriz de similitudes, obtendremos el *minimum spanning tree***.** En la representación en forma de árbol es mucho más sencillo encontrar que mejores relaciones de las comparaciones realizadas. En este ejemplo se pueden distinguir directamente mirando la matriz de similitudes. Pero si se imagina una matriz con 48 especies comparadas, se generarían más de 1000 resultados. Es por eso que resulta tan útil la simplificación que nos proporciona la construcción del árbol.

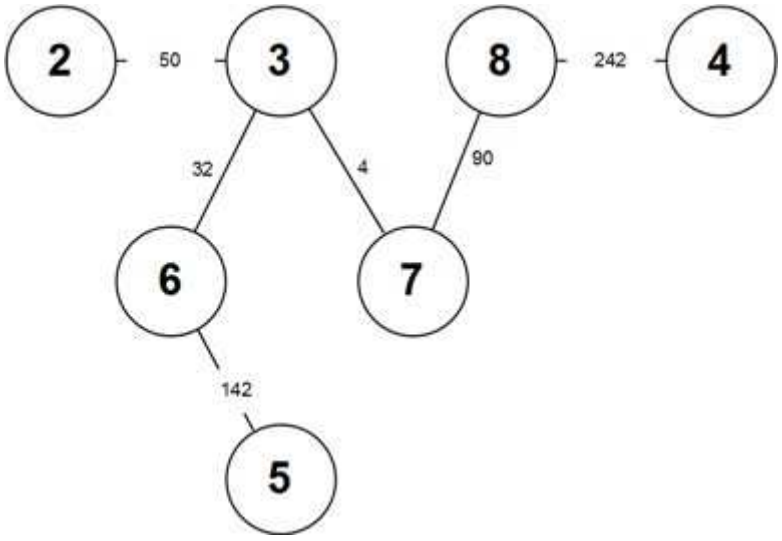


Figura 24 Minimum spanning Tree Construido a partir de la matriz de similitudes

5. Informe Técnico

En esta sección se detallara los ficheros, carpetas y programas utilizados durante todas las fases del proyecto. Es importante tener en cuenta que todas las herramientas desarrolladas en el proyecto estarán automatizadas, sin que se necesite intervención humana, por tanto si los programas no están colocados correctamente en la estructura de carpetas del servidor, el sistema no funcionara correctamente.

5.1 Estructura de Archivos del Proyecto

A continuación se listaran la estructura de carpetas utilizada para utilizar y almacenar los ficheros y programas de todas las fases del proyecto.

Carpeta	Ficheros / Carpeta	Descripción
/Archaea	/mums	Carpeta donde se almacenan los MUMs de Archaea
	/smums	Carpeta donde se almacenan los superMUMs de Archaea
	genes.txt	Fichero donde se almacenan los nombres de las especies substituidos por su identificador
	factors.txt	Fichero donde se almacenan la cantidad de superMUMs obtenidos en cada comparación
	mstprim.txt	Fichero con el *minimum spanning tree* resultante
/Bacteria	En estas dos carpetas se repite la misma disposición de ficheros y subcarpetas que en Archaea	
/Eukaryota		
/factores2samples	/Datasets	Carpeta donde se guardan los ficheros resultantes del programa
	factors2samples.cc	Código del programa de búsqueda de mejores comparaciones
/lanza_mums	/Mumunix	Carpeta donde está el programa para el cálculo de MUMs
	/CalculoSmums	Carpeta donde está el programa para el cálculo de superMUMs
	/CalculoSmums/smum.cc	Programa para el cálculo de superMUMs
	lanza_mums.cc	Programa lanzadera que automatiza el cálculo de MUMs
	lanza_smums.cc	Programa lanzadera que automatiza el cálculo de superMUMs
/regions	/CalculoConservacion /grado_conservacion.cpp	Fichero del programa para el cálculo del grado de conservación entre regiones de un genoma a otro.
	Lanza_grados.cpp	Programa lanzadera de la búsqueda de regiones conservadas para todos los genomas
	Calculo_regiones.cpp	Programa que busca la conservación

		de una región
/mstree	/Datagrames	Carpeta donde se generaran los ficheros de salida
	/programa	Contiene los datos del programa

5.2 Descripción y Uso de los Diferentes Programas

A continuación tendremos una lista con los diferentes programas, su uso, su llamada y la descripción de los formatos de entrada y salida.

5.2.1 Calculo de superMUMs

- smums.cc

El programa smums.cc es el encargado de aplicar el algoritmo para el cálculo de superMUM sobre un fichero de MUMs de entrada. El formato del nombre de los ficheros de MUMs debe ser el mismo parar todos, tomamos por ejemplo la comparación entre dos especies con identificadores 1 y 2 respectivamente. Los ficheros de MUMs resultantes de su comparación serian:

1_2.directo y 1_2.inverso

Donde los números son los identificadores comparados y directo e inverso representan la orientación de los MUMs comparados.

La estructura de todos los ficheros MUMs debe ser siempre la misma para que el algoritmo de superMUM funcione correctamente. Los ficheros estarán formados por líneas de tres números enteros positivos cada uno, donde cada línea representa un MUM. El primer entero es la posición inicial del MUM en el genoma de la primera especie comparada, el segundo valor es la posición inicial en el genoma de la segunda especie y el tercer valor es el tamaño que tiene ese MUM contado en número de bases.

Poniendo un ejemplo de un fichero con N MUMs:

Fichero con N MUMs

MUM 1	Pos.Inicial1	Pos.Inicial2	TamañoMUM
MUM 2	Pos.Inicial1	Pos.Inicial2	TamañoMUM
	…	…	…
MUM N	Pos.Inicial1	Pos.Inicial2	TamañoMUM

Para el correcto funcionamiento del algoritmo de superMUMs es muy importante que los MUMs de los ficheros estén ordenados por la posición inicial en el genoma1, de menor a mayor.

Para la compilación del programa se utiliza un conjunto de compiladores de software libre llamado GPP. Los parámetros de entrada del programa sumum.cc son el nombre del fichero de MUMs deseado y un número que representa el multiplicador (Por defecto se utiliza un multiplicador de 2.5).

Compilación: **gpp smum.cc –o smum**
Ejecución: **./smum NombreFicheroMUMs Multiplicador**

Los ficheros de superMUMs resultantes al cálculo tendrán la misma estructura que los ficheros de MUMs. Pero como deben incluir tanto los superMUMs de la comparación como los MUMs no absorbidos se añadirá una línea en blanco para diferenciarlos dentro del fichero.

Fichero con N superMUMs y M MUMs

superMUM 1	Pos.Inicial1	Pos.Inicial2	Tamaño_superMUM
superMUM 2	Pos.Inicial1	Pos.Inicial2	Tamaño_superMUM
…	…	…	…
superMUM N	Pos.Inicial1	Pos.Inicial2	Tamaño_superMUM
	Línea divisoria en blanco		
MUM 1	Pos.Inicial1	Pos.Inicial2	TamañoMUM
MUM 2	Pos.Inicial1	Pos.Inicial2	TamañoMUM
…	…	…	…
MUM M	Pos.Inicial1	Pos.Inicial2	TamañoMUM

El programa smum.cc también generara un archivo de estadísticas para comparación, donde simplemente será una línea con el nombre del fichero MUM de entrada, el número de superMUMs, la suma del tamaño total de estos y el número de MUMs no absorbidos.

- lanza_smum.cc

El programa lanza_mums.cc es el encargado de automatizar el cálculo de los superMUMs. Se le pasara el nombre de la carpeta donde están almacenados los todos los ficheros de MUMs a comparar y el programa se encargara de leerlos todos y llamar a smum.cc con cada uno de ellos. Opcionalmente se le puede decir que multiplicador deseamos que use en el cálculo de superMUMs, por defecto será 2.5.
La compilación también se realiza con gpp.

Compilación; **gpp lanza_smums.cc –o lanza_smums**
Ejecución: **. /lanza_smums NombreCarpetaMUMs [Opcionalmente: Multiplicador]**

Finalmente el programa se encargara de unir en un mismo fichero todas las estadísticas generadas durante los cálculos, independientemente de si son directos o inversos. Sumará el número de superMUMs resultantes de todos los cálculos y el tamaño de todos ello. Este archivo recibirá el nombre de **factors.txt.** Un ejemplo de fichero de factors.txt seria el siguiente.

Fichero factors.txt con 3 genomas (A,B,C)

GenomaA	GenomaB	SumaTamañoSuperMUMsAB
GenomaA	GenomaC	SumaTamañoSuperMUMsAC
GenomaB	GenomaC	SumaTamañoSuperMUMsBC

- factors2samples

Este programa nos permitirá saber cuáles son las mejores relaciones basándose en el fichero **factors.txt** generado por el programa de lanza_smums.cc. Se considera como una mejor relación aquella que tiene la mayor suma de tamaños de superMUMs.

Debido a que es un programa con múltiples funcionalidades se le debe especificar qué tipo de salida se espera al llamarlo. Para los fines de nuestro proyecto utilizaremos los parámetros de entrada **-a** y **–d**, que significan que deseamos una salida con todas las especies relacionas ordenadas de mayor a menor.

El resto de los parámetros de entrada son, el número de especies que se han comparado en el cálculo de superMUMs y el fichero de **factors.txt**. La compilación se realiza con gpp.

Compilación; **gpp factors2samples.cc –o factors2samples**
Ejecución: **. /factors2samples NumeroEspecies factors.txt –a –d**

El resultado del programa es una lista con los mismos parámetros que **factors.txt** pero ordenada por el tamaño total de los superMUMs. De esta manera se sabrá cual ha sido la mejor relación.

5.2.2 Búsqueda de Regiones Conservadas

- grado_conservacion.cpp

Es el programa encargado de la búsqueda de una región conservada de un genoma a otro genoma. Para ello se necesitan los ficheros de superMUMs tanto directos como inversos de la comparación de ambos genomas.

El resto de los argumentos de entrada será la posición inicial y final de la región en el primer genoma, un multiplicador para el agrupamiento y eliminación de los superMUMs, una marca que indicara si tenemos que invertir las posiciones de los superMUMs de los archivos (Ahora se explicara el porqué) y finalmente el nombre del archivo de salida al que se le añadirá la extensión **.grados**.

La *flag* que indica la inversión de las posiciones de los superMUMs es necesaria porque los superMUMs están ordenados por las posiciones del primer genoma, pero si la

especie de la cual queremos obtener la conservación es la segunda se deben intercambiar las posiciones del primer genoma con el segundo para que el resto del programa funcione correctamente.

Compilación; **gpp grado_conservacion.cc –o grado_conservacion**
Ejecución: **. /grado_conservacion ficheroSuperMUMDirectos superMUMsInversos InicioRegion FinRegion Multiplicador FlagInversion NombreFicheroSalida**

El fichero de salida mostrara el grado de conservación obtenido y la región del segundo genoma correspondiente a ese grado. Un ejemplo podría ser:

Nombre_Fichero	GradoConservacion	InicioRegion	FinRegion

- Lanza_grados.cpp

Este programa se encargara de hacer las búsquedas de conservación de una región para todos los especies. El programa solo necesitara el identificador de la especie deseada, el dominio de esta y la región en donde se hará la búsqueda. También se debe especificar que multiplicador se aplicara en el cálculo de los grados.

Este programa aplicara el calculo_grado.cpp con los ficheros superMUMs de las comparaciones entre el genoma especificado por parámetro y el resto de especies del dominio utilizando siempre la misma región para todos los cálculos.

Compilación; **gpp lanza_grados.cpp –o lanza_grados**
Ejecución: **./lanza_grados Dominio(Archaea/Bacteria/Eukaryota) GenomaBase InicioRegion FinRegion Multiplicador**

Finalmente unirá todos los resultados en un mismo fichero y los ordenara por grado de conservación de mejor a menor. El nombre del fichero de salida debe incluir el identificador del genoma y la región utilizada en el cálculo. Por ejemplo, si se ejecutara el programa con el genoma 1 y la región 1 – 1000 el fichero resultante sería **1_1_1000.grados**.

- Calculo_regiones.cpp

Con este programa se realiza la búsqueda de conservación entre las regiones de todas las especies almacenadas en uno de los ficheros de **.grados** del programa lanza_grados. Necesitará como argumentos el dominio de las especies incluidas en el archivo, el identificador de genoma y la región deseada. Así podrá buscar si existe algún archivo de grados con esas características.

Compilación; **gpp calculo_regiones.cpp –o calculo_regiones**
Ejecución: **./calculo_regiones Dominio(Archaea/Bacteria/Eukaryota) IdentificadorGenoma InicioRegion FinRegion**

El fichero de salida del programa será una matriz de distancia con los grados de conservación obtenidos entre todas las especies. El nombre del fichero seguirá siendo el identificador más las regiones utilizadas pero con la extensión de **.matriz**. La estructura del fichero serán líneas con cada operación realizada, en cada línea están los identificadores de los dos genomas y el grado de conservación que hay entre ellos.

Fichero ".matriz" con 3 genomas (A,B,C)

GenomaA	GenomaB	GradoAB
GenomaA	GenomaC	GradoAC
GenomaB	GenomaC	GradoBC

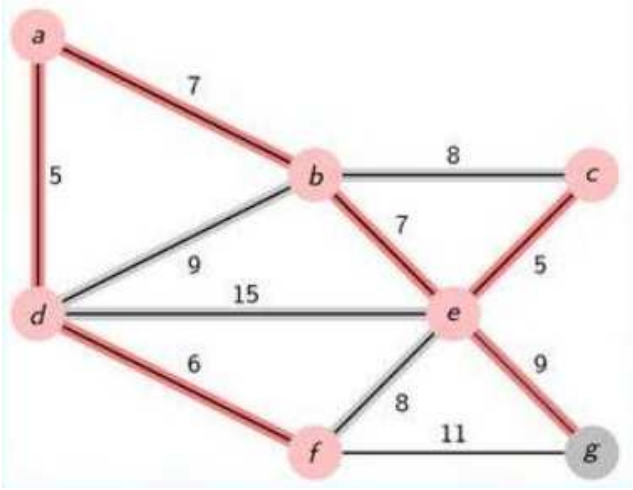
5.2.3 Construccion del Arbol *Minimum Spanning Tree*

El último programa se encarga de crear un árbol *minimum spanning tree* utilizando alguno de los ficheros **.matrices** producidos por el programa calculo_regiones.cpp. El programa no acepta ficheros por parámetros, así que se debe dejar una copia del fichero con la matriz en la carpeta /Datagrames y cambiarle el nombre por **factors.txt**.

Compilar: **gcc prim.c –o prim**
Ejecución: **./prim 1**

La salida del programa tendrá como nombre **mstree.txt** y estará formado por los nodos y el valor de estos que se han elegido por el algoritmo del árbol.

El ejemplo de un fichero mstree.txt para un árbol concreto seria:



MINIMUM SPANNIG TREE

2____7.000000____1
3____5.000000____5
4____5.000000____1
5____7.000000____2
6____6.000000____4
7____9.000000____5

6. Conclusiones

Aplicando la metodología desarrollada, que se muestra en el apartado 3, se han podido cumplir los objetivos del proyecto, ya que se han podido crear todas las herramientas necesarias para la realización de los mismos.

La parte del pre-proceso que consistía en crear una nueva herramienta de comparación útil, se ha cumplido satisfactoriamente. Los superMUMs son funcionales y eficientes. Por ejemplo en una comparación entre la especie humana y macaco, con más de 83 millones de MUMs generados por la comparación, el cálculo de superMUMs no supera los 15 minutos.

La parte de trabajo online que consistía en la búsqueda de regiones conservadas en diferentes genomas para obtener los posibles ancestros comunes entre especies, también ha sido completado y se han podido crear árboles que representan los mejores niveles de conservación de cierta región del genoma de una especie en el resto de las especies.

Las herramientas desarrolladas pueden ser muy interesantes para la comunidad científica, con solo crear una aplicación web que utilice los programas de búsqueda de regiones, se podrá hacer la búsqueda de ancestros comunes totalmente on-line. Esto facilitara mucho las cosas para todos aquellos investigadores interesados en genómica comparativa.

Personalmente me siento muy satisfecho con el trabajo realizado y los objetivos alcanzados. Aunque en algunas ocasiones ha sido complicado, sobretodo el diseño e implementación del algoritmo de superMUMs, ha sido muy estimulante conocer aplicaciones de la informática en el campo de la medicina, en este caso en el estudio de la genética. Eso me ha llevado a tener en cuenta otras posibles alternativas en mi futuro profesional.

La parte que más me ha gustado ha sido la realización de las matrices de similitud y la aplicación del algoritmo de Prim en estas para obtener los arboles resultado. Ver que todo el esfuerzo puesto en el proyecto daba sus frutos ha sido muy gratificante.

Este proyecto para mí ha sido una manera de saber que los conocimientos adquiridos durante estos años de carrera los puedo aplicar para desarrollar proyectos complejos y largos por mí mismo.

7. Referencias

1. A.L. Delcher, S. Kasif, R.D. Fleischmann, J. Peterson, O. White, and S.L. Salzberg, **Alignment of Whole Genomes**. Nucleic Acids Research, 27:11 (1999), 2369- 2376.

2. Huerta,M. and Messeguer,X. **Efficient space and time multicomparison of genomes**. Research Report LSI-02-64-R, Dep. Llenguatge i Sistemes Informàtics, Universitat Politècnica de Catalunya (2002).

3. Ferre D, Roset R, Huerta M, Adsuara JE, Rosello L, Alba MM, Messeguer X**: Identification of patterns in biological sequences at the ALGGEN server: PROMO and MALGEN**. Nucleic Acids Res 2003, 31(13):3651-3653.

4. Treangen T. and Messeguer X. **M-GCAT: Interactively and efficiently constructing large-scale multiple genome comparison frameworks in closely related species**. BMC Bioinformatics 2006, 7:433.

5. Mario Huerta **Suffix Tree Construction with slide nodes.** technical report LSI-02-63-R Dep. Llenguatge i Sistemes Informàtics, Universitat Politècnica de Catalunya (2002).

6. http://platypus.uab.es : Web server for the all-known-genomes comparison by web. Server supported by the Institute of Biotechnology and Biomedicine of the Autonomous University of Barcelona (IBB-UAB).

8. Resumen

Resum

La cerca de similituds entre regions de diferents genomes ofereix molta informació sobre les relaciones entre les especies d'aquest genomes. Es molt útil per a l'estudi de la conservació de gens d'una especia a un altre, de com les propietats d'un gen son assignades a un altre gen o de com es creen variacions en genomes diferents durant l'evolució d'aquestes especies.
La finalitat d'aquest projecte es la creació d'una eina per a la cerca d'ancestres comuns de diferents especies basada en la comparació de la conservació entre regions dels genomes d'aquestes especies.
Per a una comparació entre genomes mes eficaç una part important del projecte es destinarà a la creació d'una nova unitat de comparació. Aquestes noves unitats seran superestructures basades en agrupació dels MUMs existent per la mateixa comparació que anomenarem superMUMs. La aplicació final estarà disponible al servidor: http://revolutionresearch.uab.es

Resumen

La búsqueda de similitudes entre regiones de diferentes genomas aporta mucha información sobre las relaciones entre las especies de estos genomas. Es muy útil para el estudio de la conservación de genes de una especie a otra, de cómo las propiedades de un gen son asignados a otro gen o de cómo se crean variaciones en genomas diferentes durante la evolución de esas especies.
La finalidad de este proyecto es la creación de herramientas para la búsqueda de ancestros comunes de distintas especies basada en la comparación de la conservación entre regiones de los genomas de dichas especies.
Para una comparación entre genomas más eficaz una importante del parte del proyecto se dedicara a la creación de una nueva unidad de comparación. Estas nuevas unidades serán superestructuras basadas en agrupaciones de los MUMs existentes de la misma comparación que llamaremos superMUMs. La aplicación final estará disponible en el servidor: http://revolutionresearch.uab.es

Abstract

The search for similarities between regions of different genomes provides much information on the relationships among species of these genomes. As the study of conservation of genes

from one species to another, how the properties of a gene are assigned to another gene or how to create variations in different genomes during the evolution of these species.
This project describes the creation of tools to search for common ancestry of different species based on a comparison of conservation among regions of the genomes of these species
For a more effective comparison of genomes a part of the project will be dedicated to the creation of a new unit of comparison. These new units will be superstructures based on groupings of existing MUMS called superMUMs. The final application will be available at the server: http://revolutionresearch.uab.es