Servidor web para la comparación de genomas y creación del árbol filogenético

Iván Varón

Resumen– Los seres vivos compartimos parte de nuestro código genético entre distintas especies. Estudiar la similitud del código genético del ser humano con otras especies animales permite extrapolar conclusiones de experimentos con animales, e incluso conocer el carácter evolutivo de una enfermedad. Este proyecto ayuda a realizar este tipo de investigación ofreciendo una aplicación web que permite comparar el código genético de los genomas de todos los seres vivos. En la aplicación web se pueden comparar genomas de eucariota, bacterias y arquea. La aplicación web muestra un árbol filogenético para cada tipo de organismos. Este árbol representa visualmente la similitud entre distintas especies de eucariotas, bacterias y arqueas, así como su ascendencia común. Posteriormente, una vez seleccionados en el árbol los genomas que se quieren comparar, el visor en detalle permite ver qué partes de un genoma se conservan en el resto de genomas.

Palabras clave– Árbol filogenético, Arquea, Bacteria, Bioinformática, Eucariota, Gen, Genoma, MUM, NCBI, SMUM.

Abstract– We, living creatures, share out genetic code between species. Studying the similitude between the genetic code of the human being and other animal species makes it easier to extract conclusions from animal testing, and grants information about the origin of some diseases. This project helps this type of investigation offering a web application that allow for the comparison of the genomes of all living creatures. In this web application, genomes of eukaryote, bacteria and archaea can be compared. The web application shows a phylogenetic tree for each type of organism. This phylogenetic tree visually represents the similitude between species of eukaryote, bacteria and archaea, and also shows their shared ancestors. The genomes in the tree can be selected to open the detailed view of the comparison between the selected genomes. The detailed view allows for the viewing of parts of the genomes that are conserved in other genomes.

Keywords– Archaea, Bacteria, Bioinformatics, Eukaryota, Gene, Genoma, MUM, NCBI, Phylogenetic tree, SMUM.

1 Introducción y estado del arte

El genoma de todo ser vivo contiene su información genética. Los genomas están formados por largas secuencias de cuatro bases nitrogenadas (A, C, T, G). Todos los seres vivos poseemos genes en nuestro ADN. Estos genes contienen la información codificada de cualquier función fisiológica. Nosotros, los seres humanos, compartimos genes con muchos otros seres vivos.

La comparación de genomas proporciona mucha información sobre los procesos evolutivos que han experimentado los seres vivos que hoy día viven en la Tierra. Además, esta comparación de genomas tiene una aplicación médica muy importante. Actualmente, la asignación de funciones para nuevos genes se realiza mediante el reconocimiento de subsecuencias funcionales de un organismo a otro.

Podemos clasificar los genomas en tres tipos: arqueas, bacterias y eucariotas. Los genomas de eucariotas son de un tamaño mucho mayor a los de arqueas y bacterias, y esto supone un problema para su comparación. Por ejemplo, el genoma humano tiene tres mil millones de bases, mientras que el genoma de una bacteria tiene alrededor de tres millones.

La bioinformática hace posible encontrar similitudes entre el código genético de distintas especies. La cantidad de datos a analizar es muy grande y sin ayuda de la informática sería prácticamente imposible.

Este proyecto pretende hacer más fácil la investigación representando visualmente las zonas secuencias genéticas compartidas, además de representar la similitud entre las

• E-mail de contacto: ivan.varon@e-campus.uab.cat
• Mención realizada: Computación
• Trabajo tutorizado por: Mario Huerta, Francisco Javier Sánchez
• Curso 2016/17

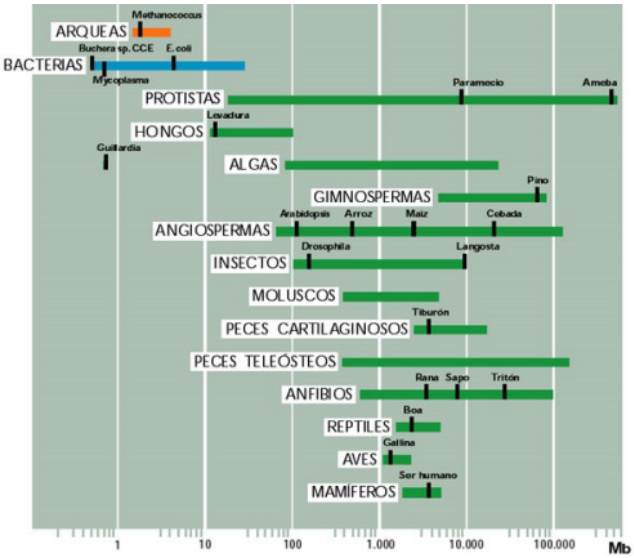

Fig. 1: Diferencias entre el tamaño de los genomas para cada gran grupo de genomas, medido en millones de bases. En naranja las arqueas, en azul las bacterias y en verde las eucariotas.

distintas especies con un árbol filogenético que nos muestra su origen común.

Existen diversos métodos para comparar genomas. La técnica utilizada por la línea de investigación del IBB [1] para la comparación de genomas completos se basa en el cálculo de los Maximal Unique Matchings (MUMs) entre las secuencias de los genomas comparados [2]. Los MUMs son las subsecuencias de bases nitrogenadas comunes y únicas de mayor longitud entre los genomas comparados. En la figura 2 podemos ver un ejemplo de subsecuencia común entre dos secuencias, que sería un MUM.

ACTGACTTAGAT**GATGACCA**TAGTAATGCGCGATCGATCGATCGTA

CGTCAGTCATGCATGCTATATACGATCGA**GATGACCA**ACTGACGAT

Fig. 2: Dos secuencias con una subsecuencia común de tipo MUM

Las subsecuencias de genomas pueden estar codificadas de izquierda a derecha o de derecha a izquierda. De esta manera se pueden encontrar fragmentos del código genético que durante su proceso evolutivo invierten su orden de una especie a otra manteniendo la funcionalidad. Así, podemos clasificar los MUMs en directos (se ha mantenido el orden) o inversos (se ha invertido el orden).

A partir de los MUMs se calculan los SuperMUMs (SMUMs) que permiten tratar genomas muy grandes como los eucariotas con más facilidad. Un SuperMUM es una agrupación de MUMs consecutivos con pequeñas subsecuencias diferentes entre ellos (approximate string matching).

Una estructura de datos utilizada para obtener los MUMs entre dos genomas son los suffix trees [3]. El algoritmo MUMS-OL utiliza suffix tres para obtener los MUMs mientras lee el genoma a comparar sobre el suffix tree del otro genoma comparado [4]. Este algoritmo es utilizado por servidores de aplicaciones que comparan secuencias genómicas [5]. El servidor del IBB [6] utiliza el algoritmo MUMOL para realizar el cálculo de los MUMs.

Todos los genomas secuencializados hasta el momento se pueden obtener accediendo al servidor del National Center for Biotechnology Information (NCBI) [7] del gobierno de los EEUU. Este mismo servidor contiene los genes de cada genoma con las posiciones que ocupa cada gen dentro de la secuencia génica completa.

Para representar las dependencias evolutivas entre los genomas de diferentes especies se utilizan los árboles filogenéticos. Estos árboles muestran la evolución entre diversas especies con ascendencia común. Los árboles filogenéticos suelen basarse en rasgos conservados entre las especies, pero proporcionan resultados mucho más precisos cuando utilizan la información obtenida a partir de las comparaciones entre los genomas de los diversos organismos.

El IBB [1] sigue una línea de investigación para la comparación de genomas enteros. Para este propósito se está trabajando para ofrecer un servidor público (http://www.platypus.uab.cat/) que permita consultar vía web las comparaciones entre genomas y su árbol evolutivo.

Para poder construir los arboles filogenéticos basados en las comparaciones de genomas son necesarios unos cálculos previos que nos permitan obtener todos los datos necesarios sobre la similitud entre los genomas.

En la línea de investigación del IBB [1], denominamos preproceso a todos los cálculos previos a la ejecución de la aplicación web, de forma que cuando el usuario final utilice esta aplicación web, todos los cálculos de comparación entre genomas ya estén hechos, y lo único que tenga que hacer la aplicación sea ir a buscar los ficheros resultantes de los cálculos del preproceso.

En la línea de investigación del IBB, el preproceso se divide en dos fases. La primera fase corresponde a la descarga de genomas desde el servidor del NCBI [7], el cálculo de los MUMs y los SMUMs entre todos los genomas y, a partir de estos cálculos, la creación de la matriz de similitud entre genomas. La matriz de similitud almacenará entonces la similitud entre cada par de genomas.

La similitud entre un par de genomas se obtiene a partir de la suma de los MUMs o SMUMs resultantes tanto en dirección directa como en inversa.

La segunda fase corresponde a la creación del árbol filogenético entre los genomas comparados, la creación de los clusters con los genomas más similares y el layout (disposición en el plano 2D) del árbol filogenético. Posteriormente será mostrado en el aplicativo web.

Uno de los principales problemas es que la primera fase del preproceso (cálculo de MUMs y SMUMs) es muy costosa en tiempo de ejecución. Principalmente debido al gran tamaño de los genomas de eucariota. El cálculo de una comparación entre dos genomas de eucariota puede durar varios días utilizando el total de recursos de la máquina. En el caso de las bacterias los genomas son de un tamaño mucho menor que los genomas de eucariota, pero existen más de 3000 genomas de bacteria secuenciados lo cual implica que la fase 1 del preproceso sea también muy lenta para los genomas de bacteria.

El principal factor que aumenta el coste temporal de la fase 1 del preproceso es que los genomas deben compararse todos contra todos, y en inversa y directa. Además, los suffixtree que se crean para buscar los MUMs en geno-

mas tan grandes como los de Eucariota pueden colapsar la memoria RAM. Por otro lado, otro problema importante es el almacenamiento de los resultados de las comparaciones. En el caso de la comparación de bacterias se generan miles de ficheros de resultados por lo que se podría sobrepasar la capacidad de almacenamiento del servidor.

En la segunda fase del preproceso se trabaja a partir de la matriz de similitud entre genomas por lo que las limitaciones de almacenamiento de resultados ya no son un problema y su tiempo de cálculo es reducido.

La descarga de genomas del NCBI debería ser automática para que la base de datos local del aplicativo, y el árbol filogenético obtenido, esté siempre actualizada. Este problema se tratará en este trabajo.

En el servidor del IBB [6] disponemos de una interfaz gráfica web, bautizada como Mummy, que permite la visualización de las comparaciones entre múltiples genomas en detalle. Esta interfaz gráfica permite estudiar y analizar en detalle los MUMs y SMUMs entre pares de genomas.

En el servidor del IBB [6] también dispone de una interfaz web diferente que muestra el árbol filogenético entre todos los genomas comparados.

Actualmente para la interfaz Mummy es necesario almacenar los MUMs y SMUMs de cada comparación para la muestra de resultados. Tal y como se ha comentado anteriormente el almacenamiento de todos los MUMs entre los genomas comparados implica un gran volumen de ficheros lo cual no es viable.

2 OBJETIVOS

- Testeo del robot de descarga de genomas de eucariota e implementación del robot de descarga de genomas de bacteria:

  El robot de descarga de genomas de eucariota ya está programado y yo debo asegurarme de que funcione bien con todos los casos que se puedan dar: genomas nuevos (descargar) y genomas que ya tenemos (si han sido modificados, descargar). El robot de descarga para bacterias y archeas ha de desarrollarse.

- Testeo de lanzar la fase 2 tras comparar un genoma de bacteria con los anteriores e implementación de lanzar la fase 2 tras comparar un genoma con los anteriores para eucariotas:

  En la implementación de la fase 1 de bacterias ya llama correctamente a la fase 2 cada vez que se compara un nuevo genoma con el resto con tal de incorporarlo al árbol. Yo tengo que asegurarme de que el proceso se hace correctamente. El lanzamiento de la fase 2 para Eukariotas ha de desarrollarse.

- Testeo del filtro de genomas de bacteria y genomas de arquea para especies de la mima familia:

  En vez de añadir todos los genomas de bacterias y arqueas disponibles, nos quedamos solo con un genoma representativo de cada familia de genomas. Esto mejora mucho el árbol filogenético porque las bacterias y arqueas de la misma familia son muy similiares y la información que se puede extraer es prácticamente la misma. Tengo que adaptar el filtro a la descarga automática de genomas.

- Cálculo dinámico de la comparación de genomas de bacteria y arquea:

  Las comparaciones de la fase 1 del preproceso en bacteria y arquea solo se realizan para obtener el grado de similitud entre los genomas, tras ello los archivos son eliminados. Almacenar todas las comparaciones entre todos los genomas sería muy costoso. Tengo que generar los ficheros por petición desde el aplicativo web en el momento en el que el usuario selecciona los genomas a comparar en el árbol filogenético.

- Automatización del robot que descarga genomas de la base de datos del NCBI

  Una vez implementados los robots de descarga de genomas y comprobar que funcionan correctamente, se dejarán ambos automatizados para que periódicamente descarguen los nuevos genomas secuenciados y los añadan al árbol filogenético, y para que actualicen el árbol cuando la secuencia genética o el mapeado de genes de un genoma que ya está en el árbol haya sido modificados en la base de datos del NCBI.

- Generación de los ficheros depurados para la interfaz gráfica MUMMY en bacteria, archea y Eukariota.

  Generar filtros y postratamientos de los ficheros de SMUMs y MUMs, con diferentes objetivos: Mostrar en cada momento la estructura conservada de un genoma a otro (verdaderos positivos), eliminar información redundante o ruido (falsos positivos), no dejar de mostrar información relevante (falsos negativos), no colapsar el navegador al mostrar los gráficos, minimizar el tiempo de carga de los ficheros en la computadora del usuario. Los anteriores puntos son especialmente delicados en el caso de la comparación de genomas de Eukaryota.

3 METODOLOGÍA

En esta sección voy a explicar los procedimiento que he seguido para obtener los resultados, los programas hechos y añadidos por mí y las mejoras a programas ya existentes.

3.1 Cálculo de comparación de genomas de bacteria y arquea (fase 1)

La fase 1 de bacterias y arqueas es la encargada de la comparación de genomas de bacterias y arqueas. Se encuentra en el archivo lanza_mums.cc. He aplicado ciertos cambios a la versión anterior para mejorar su funcionamiento.

He añadido un parámetro para distinguir entre arqueas o bacterias. Como la fase 1 es igual para ambos tipos de genomas, se utiliza el mismo programa, pero no se podía distinguir entre arqueas o bacterias. Otro problema resuelto es como recibía la fase 1 los identificadores de los genomas a añadir o actualizar. Estos identificadores se recibían por parámetros. Al poderse recibir miles se sobrepasaban los límites de tamaño de línea de las llamadas a programa, ahora se reciben por fichero. A la hora de tratar estos identificadores, he desarrollado un sistema para distinguir entre los diferentes tipos de comparaciones que han de realizarse: Genomas nuevos, o genomas modificados. Para ello me

baso en el orden de los identificadores de los genomas a comparar con el resto. Primero coloco los identificadores de genomas nuevos. Después los identificadores de los genomas modificados en el servidor remoto (actualizaciones). He decidido que este orden es el que mejor se adapta a lo que es más interesante para el usuario de la aplicación, es decir, tener en el árbol evolutivo el máximo de genomas en todo momento, incorporando primero los nuevos genomas al árbol, y posteriormente modificando el árbol en base a los cambios en las secuencias de los genomas. Hay que tener en cuenta que comparar un genoma de eucariota con el resto puede durar hasta 3 semanas. En esta misma línea en la función encargada de la creación de la matriz de similitud entre genomas he añadido la posibilidad de que se actualice la matriz. Esta funcionalidad es necesaria para tratar los genomas modificados (genomas que ya están en la matriz y se ha descargado una versión más reciente). Ahora, en los casos de actualización, se eliminan todas las comparaciones de la matriz con el genoma a actualizar, y se reemplazan con los nuevos grados de similitud obtenidos después de recalcular todas las comparaciones entre el genoma actualizado y el resto.

El sistema se ejecuta en paralelo para genomas de bacteria y para genomas de eucariota. A su vez se lanzan en paralelo todas las comparaciones de genomas de bacteria posibles hasta ocupar toda la cpu, y se lanzan todas las comparaciones entre los cromosomas de un par de genomas eucariota hasta ocupar todo el espacio RAM disponible. En este proceso de paralelización también se han realizado diferentes mejoras.

En genomas de bacteria se ha paralelizado la creación de los ficheros que contienen los offsets o subsecuencias todavía no codificadas de los respectivos genomas, y que por lo tanto no han de compararse pero si tenerse en cuenta a la hora de mostrar los resultados finales y el mapeado de genes. Cada comparación crea un archivo temporal con un nombre único.

Tanto para bacteria como para eucariota, he añadido un control antes de lanzar la fase 2 (el cálculo el árbol filogenético) tras cada genoma comparado con el resto, que controla si ya hay una fase 2 en ejecución. Este caso puede darse porque la fase 1 de genomas de eucariotas se lanza al mismo tiempo que la de bacterias/arqueas.

También se ha añadido un control sobre los procesos zombis generados por las múltiples ejecuciones en paralelo. Se ha añadido un bucle que se lanza tras comparar un genoma con el resto y antes de lanzar la fase 2. De esta forma el proceso padre se espera a todos los procesos hijo lanzados. Al proceso que lanza procesos lo llamamos padre, y a los procesos lanzados, hijos. Los procesos hijos los utilizamos para generar las comparaciones de genomas de forma paralela. El problema es que, en Linux, si el proceso padre no espera a que los procesos hijo terminen con la función wait, los procesos hijo se quedan en un estado llamado zombi hasta que el padre ejecute la función wait o el padre acabe la ejecución. En estado zombi, los procesos no consumen recursos de CPU, pero si mantienen su identificador de proceso (PID). El número de PID que puede asignar el sistema es finito, así que si se acumulan muchos miles de procesos zombi, como en el caso de las comparaciones de bacteria, pueden llegar a mantener tantos PID ocupados como para que no se puedan seguir lanzando procesos. Por esta razón se limpian todos los procesos zombi tras cada comparación de un genoma con el resto.

Toda la fase 1 ha sido probada para genomas nuevos, genomas modificados, y combinaciones de ambos para una gran variedad de genomas (más de 3000).

3.2 Cálculo de comparación de genomas de eucariota (fase 1)

La fase 1 de eucariota, que se encarga de generar los archivos de comparación de genomas y crear la matriz de similitud, corresponde al archivo actualiza_mums.cc. Muchos de los cambios realizados a este programa son los mismos que se realizaron en la fase 1 de bacterias. Las pruebas de todos los cambios en la comparación de eucariota se han realizado antes en la comparación de bacterias y arqueas dado que estamos hablando de 10000 veces más rápido en tiempo de cálculo.

Anteriormente vimos las mejoras aplicadas tanto a bacterias como a eucariotas, pero dado la complejidad de las comparaciones de eucariotas, estas necesitan un tratamiento especial y unas mejoras específicas. Cada lanzamiento de comparaciones de genomas se hace en paralelo, para aprovechar al máximo la capacidad del ordenador y acabar antes las comparaciones, tras ello se comprueba que la comparación haya terminado con éxito, y en caso negativo, se lanza la comparación pendiente de forma secuencial. Para controlar que las comparaciones se han realizado correctamente, se crea un fichero vacío con el siguiente formato: IDGenoma1_IDGenoma2_d.ok, donde la "d" significa directo y se pone una "i" si es inverso. Este archivo se crea al final de una comparación si el archivo de comparación de genomas se ha creado sin problemas. Para las comparaciones que no hayan acabado correctamente, este archivo no se crea. Todas las comparaciones que no tengan el archivo .ok son lanzadas de nuevo secuencialmente para asegurar que se terminan sin problemas. El control de que todos los archivos .ok existan se lanza 3 veces. También se realizaron las pruebas pertinentes y se comprobó que efectivamente se volvían a lanzar las comparaciones fallidas secuencialmente.

3.3 Robot de descargas de genomas

El robot de descarga de genomas de eucariota ya estaba en un principio desarrollado. Mi tarea era testear su funcionamiento e implementar el robot de descarga de genomas de bacteria/arquea. Pero hubo un cambio en el servidor del NCBI, el servidor de donde se descargan los genomas. Los archivos de genomas antes se encontraban en carpetas distintas, y además el formato de los archivos también era distinto. Antes, cada secuencia de genoma estaba en un archivo separado del resto de información, como por ejemplo los genes, los cuales también utilizamos para mostrar la posición de cada gen en la vista en detalle de comparación de genomas (interfaz web MUMMY). Ahora se encuentra toda la información del genoma en un mismo archivo en el servidor remoto. Para los genomas de eucariota supuso un cambio muy grande en la forma en la que se parseaban los archivos descargados, porque ahora toda la información de cada cromosoma (secuencia, genes, etc.) está en el mismo archivo.

Por ese motivo se ha rehecho todo el robot de descarga y parseo de los genomas de eucariota. Se ha mantenido la idea principal de controlar si un genoma ya está añadido al árbol o no. Si no lo está, se descarga. Si lo está, se comprueba si fue modificado en el servidor remoto tras la última descarga. Este control se realiza con un pequeño programa en java, que recibe como parámetro la url del archivo en el servidor, y compara su fecha de modificación con la de un archivo de texto que contiene los nombres de todos los genomas descargados. El archivo de texto solo se escribe cuando se realiza una descarga, así que la fecha de modificación del archivo de texto es la fecha en la que se realizó la última descarga.

El parseo o tratamiento para el nuevo formato de los archivos descargados varía mucho entre los genomas de eucariota y los genomas de bacteria/arquea. En el caso de los genomas de bacteria/arquea, es necesario en encontrar el comienzo de la secuencia del genoma dentro del archivo. La secuencia del genoma siempre es el último apartado dentro del archivo descargado, así que encontrando el inicio de la secuencia ya podemos recortarla y copiarla en un archivo a tratar. El resto del archivo contiene, entre otras cosas, los genes, que también nos interesan. Como el resto de partes del archivo no son muy extensas, no recorto los genes y simplemente paso todo el archivo (sin la secuencia) al programa que extrae los genes y los parsea para que la aplicación web de vista en detalle de comnparación de genomas (MUMMY) los muestre correctamente. El archivo resultado del parser para obtener los genes lo llamamos mapgenes. Este parser que crea el fichero mapgenes ya es capaz de detectar los genes dentro del archivo descargado, porque se utilizan las librerías de java biojava y biojavax, que detectan el formato en el que vienen los genes en los ficheros descargados del NCBI. De esta forma obtenemos los dos archivos que nos interesan: el archivo de genoma a comparar (la secuencia), y el archivo de mapgenes (para mapear los genes sobre el genoma en la herramienta de visualización).

En el caso de los genomas de eucariota, el proceso se complica ya que los genomas están divididos en cromosomas. Los cromosomas están todos en el mismo archivo descargado, separados por la cadena "//". Cada cromosoma contiene la misma información genética que un archivo de genoma de bacterias: secuencia, genes, nombre, etc. En el caso de bacterias nos interesa la secuencia y los genes, pero en el caso de los cromosomas también nos interesa el identificador, que puede ser numérico, alfabético, o una combinación de ambos (01, X, ssa03, etc.). Además de los cromosomas, también se encuentran en el archivo de eucariota descargado pequeñas partes del genoma que aun no han sido secuenciadas, y estas se ignoran puesto que no pueden ser comparadas.

El programa encargado de parsear los ficheros de genomas de eucariotas descargados, procesoDescEuka.cc, consiste en un bucle principal que busca los cromosomas dentro el archivo. Por cada separador "//" encontrado, recorta la parte del nombre del posible cromosoma. Si esa parte no contiene la palabra "chromosome", se descarta y se pasa a la siguiente subsecuencia. Si contiene la palabra "chromosome", entonces se busca la descripción de la subsecuencia, puesto que hay casos en los que se tratan de partes de un cromosoma sin secuenciar. Las palabras que usamos para descartar un posible cromosoma son "unlocalized", "unplaced", "scaffold", "plasmid" y "patch". Si contiene alguno de estos términos, se descarta el posible cromosoma y se pasa al siguiente. Si no contiene niguna de las palabras, se procesa el cromosoma.

El proceso del cromosoma es exactamente igual que el proceso de los genomas de bacteria y arquea, salvo por que también buscamos el identificador del cromosoma. Todos los cromosomas tienen, en el nombre, el identificador entre la palabra "chromosome" y una coma (","). Un ejemplo sería: "'[...]chromosome X,[...]". En este caso el identificador del cromosoma es X. En el caso de encontrar un identificador numérico, o un identificador que contenga letras y números, se extrae el valor numérico y se guarda con el número representado en dos cifras. Es decir, un número 3 se cambia por 03. Esto es necesario porque el programa que junta los cromosomas, una vez procesados todos, utiliza la ordenación de strings, y sin los 0 por delante de los números de una cifra, el orden sería erróneo.

En el caso de tratarse de un genoma actualizado (es decir, que ya se encuentra en el árbol y se ha vuelto a descargar), como los archivos de genomas que descargamos del servidor contienen toda la información (secuencias, genes, etc), que haya cambiado la fecha de modificación de un archivo no significa que haya cambiado la secuencia de alguno de los cromosomas. Pueden haber cambiado los genes, o pueden haber cambiado los no-cromosomas que evitamos al procesar los archivos de genomas de eucariota descargados. Por lo que se comprueba si hay diferencias entre la secuencia de cada cromosoma de los que se acaban de descargar con los cromosomas que ya tenemos de la antigua descarga. Si todas las secuencias de los cromosomas son exactamente iguales a la versión que se descargó anteriormente, solo se sobreescriben los mapgenes (archivos que indican a la vista en detalle de la aplicación web donde se encuentran los genes dentro del genoma). Si existe alguna diferencia entre los cromosomas, se lanzarán todas las comparaciones con la nueva versión de este genoma. Esta comprobación es muy importante ya que comparar un genoma de eucariota con todos los que ya hay en el árbol es muy costoso en tiempo, puede durar más de 3 semanas, así que evitando estas "falsas actualizaciones" ahorramos mucho tiempo.

Ambos robots de descarga, tanto el de eucariotas como el de bacterias y arqueas, tienen el mismo final de proceso. Asignan un identificador de genoma a los genomas nuevos, renombran todos los archivos (cromosomas y genoma) con este identificador y mueven los archivos a la carpeta correspondiente. En el caso de ser un genoma ya existente en el árbol que se va a actualizar, no se asigna un nuevo identificador, se mantiene el identificador que ya tenía. Todos los identificadores se escriben en el fichero comentado en la sección anterior y se lanza la comparación de genomas, que leerá este fichero para saber los genomas que tiene que comparar con todos los que ya están en el árbol, cuales son los genomas nuevos y cuales los actualizados.

3.4 Filtro de genomas de bacteria y arqueas

El filtro de genomas de bacteria y arquea consiste en quedarse con un representante de cada familia de genomas de bacteria o arquea. Este filtro se realiza porque hay muchísimos genomas de bacteria que, al ser de la misma

familia, aportan prácticamente la misma información a la comparación, y dificulta la construcción del árbol. El filtro se ejecuta en el robot de descarga de genomas de bacteria/arquea, antes de empezar a procesar los archivos de genoma descargados.

El filtro es un script de bash llamado netejaFitxers.sh. Primero ordena alfabéticamente todos los genomas descargados de bacteria. Como los genomas de bacteria y arquea siempre tienen nombres compuestos, el programa considera genomas de la misma familia a todos los genomas que empiecen por las mismas dos palabras. Por ejemplo, "Aeromonas hydrophila 4AK4" y "Aeromonas hydrophila AL09-71" son dos bacterias de la misma familia. Al estar ordenados alfabéticamente los nombres de los genomas, nos quedamos con el primer genoma de cada familia distinta que encontremos. El resto son descartados.

Para que genomas nuevos que pertenezcan a familias de las que ya tenemos un representante en el árbol no se descarguen, se extrae la familia de cada genoma (las dos primeras palabras del nombre) y se busca entre los nombres todos los genomas que están en el árbol si alguno comienza con esas dos primeras palabras. Si tenemos un genoma de la misma familia ya añadido en el árbol, no lo descargamos.

3.5 Cálculo de comparaciones de genomas de bacteria y arquea bajo petición

Cuando incorporamos la similitud entre dos genomas de bacteria a la matriz de similitud entre genomas, borramos los archivos, ya que son muchos (aprox 3000x3000 comparaciones /2). El cálculo bajo petición de la comparación de genomas de bacteria consiste entonces en generar los archivos de comparación de genomas cuando un usuario quiere ver su comparación en detalle mediante la interfaz web (MUMMY). Como el cálculo es rápido (segundos), es posible hacerlo en el momento en el que se quiera ver la comparación sin que el usuario perciba la espera. Este programa se llama dynamic_mums.cc.

Además de los MUMs, se calculan los super MUMs (SMUMs). Los SMUMs son un approximate string matching, donde múltiples MUMs (exact matching) se agrupan para formar un solo SMUM. Estos archivos los utiliza la aplicación web que muestra visualmente la comparación entre genomas cuando realiza la operación zoom out (vista alejada), mientras que los MUMs los muestra en la operación zoom in(vista de cerca).

El programa de comparación por demanda recibe por parámetro los identificadores de los genomas a comparar, y calcula los MUMs y SMUMs entre todos ellos. Una vez que el navegador del usuario recibe los archivos y puede ver la comparación en detalle de los genomas (MUMMY), no se borran los archivos de nuestro servidor. De esta forma, la próxima vez que un usuario solicite la misma comparación de genomas, los MUMs y SMUMs ya estarán calculados. Así, las comparaciones más usuales no se tienen que calcular cada vez que un usuario quiere estudiarlas.

Cada vez que el usuario escoge un grupo de genomas a comparar, se realiza una llamada al archivo dynamic_mums.php, el cual se encarga de recibir por parámetros GET los identificadores de genoma que han de compararse entre ellos. Se construye una string con todos los identificadores de genoma separados por una coma, y se realiza la llamada al archivo dynamic_mums.cc, que lanza los programas de comparación de genomas. Se envía la string de identificadores de genoma como parámetro.

El Mummy, la aplicación web de vista en detalle de la comparación de genomas, no puede mostrar más de 8 comparaciones al mismo tiempo. Si el usuario selecciona más de 8 genomas para ver su comparación en el Mummy, se dividen las llamadas en grupos de 8. Por lo tanto si un usuario selecciona 17 genomas, se crean dos grupos de 8 genomas y un genoma queda solo en una tercera llamada. Al recibir un solo genoma, el Mummy da un error, puesto que no es posible mostrar una comparación entre genomas con solo un genoma. El programa dynamic_mums.php avisará al usuario de que si quiere ver la vista en detalle de ese genoma ha de emparejarlo con otro.

3.6 Gestión del espacio del disco del servidor donde se calculan las comparaciones de genomas

Para que el servidor no colapse por que el disco duro se ha llenado, he insertado distintas instrucciones de borrado de archivos temporales en los programas del cálculo de comparaciones de genomas (lanza_mums.cc para bacterias/arqueas y actualiza_mums.cc para eucariotas). La mayoría de los archivos temporales ya se borraban el versiones anteriores, pero el borrado se realizaba al final del programa. Esto significa que hasta que no se terminaban las comparaciones de genomas, no se borraba ningún archivo, acumulando todos los archivos temporales. Estos archivos temporales pueden llegar a ser muy pesados (sobretodo en el caso de las comparaciones de genomas de eucariotas), por lo que la mejor forma de solucionar el problema es borrar cada archivo temporal en el momento en el que ya no se necesita.

En el caso de las comparaciones de genomas de eucariota, los MUMs (archivos de comparación de genomas) se crean entre los cromosomas de uno de los genomas y el otro genoma entero. De esta forma, primero se crean todos los MUMs cromosoma-genoma, y luego se juntan todos los MUMs para formar el fichero genoma-genoma completo. He añadido una instrucción al programa concatena.cc, encargado de unir los MUMs cromosoma-genoma, que borra todos los ficheros de MUMs parciales una vez ha creado el fichero de MUM completo.

El programa que calcula la compración entre un cromosoma y un genoma, mumol.cc, crea archivos temporales por cada comparación. He movido la instrucción de borrado de archivos temporales de comparación de genomas para que se realice al final de cada comparación. Este cambio se ha realizado tanto en actualiza_mums.cc (comparación de genomas de eucariota) como en lanza_mums.cc (comparación de genomas de bacteria/arquea).

Los SMUMs son creados a partir de los MUMs. Los SMUMs se necesitan para mostrar la vista más alejada (zoom out) de la comparación entre genomas en la aplicación web. La generación de los SMUMs es un proceso que se realiza hasta tres veces. La primera de las veces toma como archivo de entrada un archivo de MUMs, y junta MUMs con pequeñas gap entre uno y otro para crear grandes estructuras de MUMs. Se realizan otras dos ejecuciones del programa de generación de SMUMs, pero tomando como archivo

de entrada el resultado de la ejecución anterior (el anterior archivo de SMUMs). Esto permite juntar las estructuras de MUMs encontradas en las ejecuciones anteriores(SMUMs), y formar estructuras cada vez más grandes y con un mayor gap entre ellas. Los archivos de SMUMs de cada ejecución no se borraban hasta el final del programa, una vez se habían realizado todas las comparaciones de genomas. He añadido instrucciones que borran todos los resultados anteriores al cálculo de SMUMs que acaba de terminar. He hecho que las instrucciones de borrado sean parametrizadas (es decir, indico el nombre del archivo concreto que quiero borrar) para que no se borren archivos de SMUMs de otras comparaciones que se estén realizando en paralelo.

3.7 Creación de SMUMs

Los SMUMs son conjuntos de MUMs que forman approximate string matchings entre dos genomas. Los SMUMs son necesarios para mostrar las zonas donde hay más similitud en el Mummy, la aplicación web de vista en detalle de la comparación entre genomas. Se han realizado ciertas modificaciones en el postproceso de los de SMUMs generados para mejorar su visualización en el Mummy.

Para encontrar las zonas donde más MUMs hay entre dos genomas, en un principio siempre se realizaban tres ejecuciones de el programa de SMUMs. En la primera se juntan los MUMs para formar estructuras con muchos MUMs con pequeños gaps entre ellos, y en las dos siguientes se juntaban las estructuras generadas en la anterior ejecución para formar estructuras más grandes. Tres ejecuciones es un buen número de ejecuciones para las comparaciones entre genomas grandes que comparten muchas zonas de similitud, pero para comparaciones entre genomas pequeños o comparaciones entre genomas que no tienen grandes subsecuencias similares entre sí, no es un buen número de ejecuciones. Para solventar este problema, se comprueban el número de SMUMs creados en cada ejecución. Si el número de SMUMs creados en una ejecución es menor a 20, nos quedamos con la anterior ejecución, ya que 20 es un número muy pequeño y significa que se han juntado demasiado los SMUMs o los genomas son poco similares, hasta el punto de que los SMUMs dejan de representar zonas de gran similitud. Si esto sucede en la primera ejecución, nos quedamos de todas formas con el archivo de SMUMs de la primera ejecución, ya que se ha de tener algún archivo de SMUMs o el Mummy no podrá mostrar nada. Se han dejado 3 ejecuciones como máximo, porque en las pruebas realizadas, los SMUMs resultantes de más de tres ejecuciones no son representativos por juntarse demasiados SMUMs entre sí.

Durante las pruebas de visualización de las comparaciones se detectaron SMUMs que se solapaban (es decir, que el comienzo de un SMUM aparecía dentro de otro SMUM). Esto era un problema visual, ya que si dos zonas de mucha similitud se solapan deberían formar una sola. Además, era un problema para que los diferentes algoritmos del postproceso de MUMs filtrasen los resultados y así mostrar la información más significativa. Por lo que se creó el programa unirSMUMS.cc, que se encarga de unir todos los SMUMs que se solapan en único SMUMs que los contiene. El programa recorre el archivo de SMUMs, quedándose con el SMUM al que llamamos SMUM actual y comprobando que los siguientes no estén solapados con él. Por cada SMUM solapado con el SMUM actual, se crea un nuevo SMUM que empieza en la posición más atrasada de los dos SMUMs y acaba en la posición más avanzada de los dos SMUMs. Este nuevo SMUM se convierte en el SMUM actual. El SMUM creado a partir de fusionar un grupo de SMUMs solapados pasa a ser un único SMUM, y los solapados se eliminan de la lista.

El fichero de SMUMs creado por el programa smum.cc en cada vuelta está dividido en dos partes por una línea en blanco. Los matchings por encima de la línea en blanco son los big SMUMs, y los que están por debajo de la línea son los SMUMs y MUMs no absorbidos. Los big SMUMs son los SMUMs que se han formado en la última ejecución del programa. Los SMUMs y MUMS no absorbidos son los SMUMs y MUMs que no se han juntado con ningún otro SMUMs o MUMs en la última ejecución. Como he explicado anteriormente, el programa para obtener los SMUMS se ejecuta n veces. La primera ejecución recibe el archivo de MUMs como archivo de entrada, y las siguientes reciben como archivo de entrada el archivo de SMUMs resultado de la anterior ejecución. Por lo tanto, los MUMs no absorbidos en la primera ejecución del programa smum.cc son los MUMs que no han llegado a formar parte de ningún SMUM. Cuando tenemos el archivo de SMUMs de la última ejecución, se procesa para que no queden SMUMs dentro de otros SMUMs o SMUMs solapados, tal como he explicado anteriormente. Si esta es la última ejecución que vamos a llevar a cavo, el fichero se pasa además por el programa smumsort.sh, un script que convierte al archivo de SMUMs al formato que necesita el Mummy, la interfaz gráfica de comparación de genomas en detalle. En el fichero resultante, se añade a los big SMUMs una columna extra. Esta nueva columna indica al Mummy que esos SMUMs se han de mostrar en el zoom-out de la interfaz gráfica (la visión más alejada) y su orden en tamaño. Al abrir una comparación entre dos genomas en la aplicación web (Mummy), por defecto la vista está en modo zoom-out, así que lo primero que ve el usuario son los big SMUMs.

Los SMUMs que se hayan creado en la primera o segunda ejecución del programa smum.cc, pueden acabar como SMUMs no absorbidos en la tercera ejecución, porque no se han juntado a otro SMUM en la tercera ejecución. Por lo tanto, a la hora de crear el fichero que se le pasará al Mummy, el programa se encarga de mover los SMUMs no absorbidos tamaño mayor al SMUM más pequeño obtenido en la última ejecución encima de la línea en blanco, para que el programa los considere big SMUMs y se muestren en el zoom-out de la vista en detalle. El punto de corte a la hora de constituir el grupo de big SMUMs definitivo es el siguiente: el mayor tamaño (más restrictivo) entre el tamaño del menor SMUM de la última ejecución, y el resultado de la siguiente fórmula:

$$1.000 + \left(\frac{MinLen}{100.000}\right)$$

Donde MinLen es la longitud del genoma más pequeño de los dos que se están comparando. Se utiliza el tamaño del menor genoma porque simepre habrá big SMUMs más pequeños en comparaciones con genomas más pequeños. Por lo tanto, cuanto más pequeño sea uno de los genomas de la comparación, menos se limita el tamaño de los big SMUMs.

Finalmente se realiza un control extra, para garantizar de que hay un mínimo de big SMUMs mostrándose en el zoom out más alejado de la interfaz Mummy. Este control consiste en verificar que el sumatorio de la longitud total de big SMUMs que se pasan a la interficie supera 10 veces el punto de corte (resultado de la fórmula x 10). En caso contrario, se consideran demasiado pocos big SMUMs, y se añadirán tantos SMUMs que están bajo el punto de corte como sean necesarios para llegar a esta cifra en el total de longitud de secuencia de big SMUMS proporcionados a la interfaz. De esta forma garantizamos siempre una longitud mostrada, ya sea en bigmums de mayor tamaño o bien en muchos bigmums de un tamaño menor.

De este fichero de SMUMs que se le pasa al Mummy para proporcionar el zoom-out (la vista más alejada), también se filtran los SMUMs y MUMs no absorbidos. Estos matchings se mostrarán junto a los big SMUMs cuando se abandone el zoom out más alejado, para proporcionar un poco más de detalle a la comparación. Estos matchings se filtran para controlar que no haya una cantidad demasiado elevada de SMUMs y MUMs no absorbidos, para reducir el tiempo de carga del fichero a la computadora del usuario, no ralentizar la navegación usando la interfaz gráfica (MUMMY), y para no mostrar información que no resulte realmente relevante para el usuario. Como punto de corte se utilizará la siguiente fórmula para calcular el número de matchings a mostrar:

$$\left(\frac{MaxLen}{1.000.000}\right)$$

Donde MaxLen es la longitud del genoma más grande de los dos que se están comparando.

Con el anterior filtro nos aseguramos tener los SMUMs no absorbidos de mayor tamaño y no superar las capacidades de la computadora del usuario, así como el tiempo de carga. Pero un filtro más es añadido a los SMUMs y MUMs no absorbidos para garantizar que los matchings mostrados sean los más significativos para el usuario. Para ello descarto los SMUMS y MUMs no absorbidos que se solapan con los big SMUMs previamente filtrados en uno u otro genoma (nunca se solaparán en ambos genomas simultáneamente pues en tal caso no serían no absorbidos). De esta forma garantizo que los SMUMS y MUMs no absorbidos (por los Big SMUMS) muestren correspondencias para las zonas entre Big Smums. Es decir, por un lado tenemos las subsecuencias con gran coincidencia (los big SMUMs) y por el otro tenemos las coincidencias en las zonas de menor coincidencia (los SMUMs y MUMs no absorbidos). Los SMUMs y MUMs no absorbidos se mostrarán cuando nos acerquemos más a regiones concretas de los genomas, y los big SMUMs en la vista global.

3.8 Ficheros de MUMs para la interfaz gráfica de comparación de genomas (operación zoom in de la interfaz MUMMY

A parte de los fichero de SMUMs (contienen Big SMUMs y no absorbidos), también se le pasa a la interfaz MUMMY el fichero de MUMs. Estos MUMs se mostrarán en la operación zoom in de la interfaz MUMMY para mostrar la vista más en detalle de la comparación entre genomas. Por ese motivo ya no se muestran aproximate string matchings (SMUMs) sino exact matchings (MUMs). Sin embargo, no pueden cargarse todos los MUMs por los motivos anteriormente expuestos: tamaño de archivo, tiempo de carga, ralentizamiento de la interfaz, y dificultad en la visualización e interpretación de los resultados por parte del usuario. Por esos motivos he añadido también un postproceso para seleccionar los MUMs que proporcionarán una mayor información al usuario, evitando aquellos menos significativos. El objetivo del procedimiento seguido es mostrar solo los MUMs contenidos en los Big SMUMs de mayor tamaño, y seguir mostrando los Big SMUMs más pequeños, y los SMUMs y MUMs no absorbidos previamente mostrados en la operación zoom out. De forma que no perdemos información pero ganamos en precisión cuando reducimos la superficie de los genomas que queremos estudiar.

Este post-proceso lo realiza el programa construirMUMfinal.cc. Cuando el programa smum.cc (encargado de crear los SMUMs) realiza su primera ejecución, recibe todos los MUMs como fichero de entrada. Una vez acaba esta primera ejecución, escribe en un fichero los SMUMs que ha generado, así como todos los MUMs que no han sido fusionados en ningún SMUM (los llamados no absorbidos). Como el programa clasifica todos los MUMs que le llegan en el fichero de entrada como absorbidos o no absorbidos, procedo a que además de los no absorbidos, escriba en un nuevo fichero de salida todos los MUMs absorbidos. De esta forma tenemos todos los MUMs que forman parte de los SMUMs de la primera ejecución en un archivo distinto. Las siguientes ejecuciones del programa smum.cc (encargado de crear los SMUMs) se encargarán de unir estos SMUMs para formar SMUMs más grandes, pero los MUMs unidos que forman los SMUMs serán los que se han guardado en la primera ejecución. El programa construirMUMfinal.cc utilizará este fichero que solo contiene los MUMs que están dentro de los SMUMs formados en la primera ejecución en vez del fichero de MUMs que contiene todos los MUMs como archivo de entrada. De esta forma, construir el archivo de MUMs que usaremos para la interfaz gráfica es mucho más rápido (porque el fichero donde se buscan los MUMs dentro de los big SMUMs es mucho más pequeño en tamaño) y preciso (porque solo contiene los MUMs con pequeños gaps entre ellos que han sido usados para constituir los SMUMs).

Admás de la anterior selección de MUMs dentro de los big SMUMs a mostrar (no mostramos todos los MUMs dentro del big SMUM sino solo los que formaron los SMUMS de la primera ejecución), también filtramos los big SMUMs de los cuales mostrar sus MUMs. Los big SMUMs de gran tamaño no suponen un problema en el fichero de SMUMs, pues constituyen un único matching (una línea) dentro del fichero. Pero todos los MUMs que forman ese big SMUM son muchos, y el problema se complica. Por esa razón solo se muestran los MUMs de dentro de algunos de los big SMUMs, y consideraré 2 parámetros para obtener el punto de corte para los big SMUMs de los que voy a mostrar los MUMs: por un lado la densidad en MUMs dentro de los big SMUMs de la comparación en cuestión, y por el otro la longitud total de big SMUMs de los que hay que mostrar su contenido en MUMs. Aplicaré entonces la siguiente

fórmula:

$$CutOff = DensidadMUMS/\left(\frac{MUMCount}{MinTam}\right)$$

donde el CutOff es la longitud máxima permitida que han de sumar todos los big SMUMs de los que se mostrarán sus MUMs, densidadMUMs es una constante que marca el límite de MUMs que considero aceptable para el fichero de MUMs, MUMCount es el numero de MUMs que se encuentran en el fichero de MUMs absorbidos en la primera ejecución del programa para obtener los SMUMs, y MinTam es la longitud del genómana más pequeño de los dos genomas comparados. La constante densidadMUMs la he obtenido tras diferentes pruebas con diferentes comparaciones, hasta encontrar la densidad que me proporciona el mayor número de MUMs posible sin perder navegabilidad en la interfaz MUMMY, ni demorar mucho el tiempo de carga.

Obtengo los big SMUMs que superan el punto de corte (Cutoff) a partir del fichero de big SMUMs ordenado por longitud, y sumando la longitud de todos los big SMUMs (mayores primero) hasta llegar al límite (CutOff). El resto de big SMUMs, así como los SMUMs y MUMs no absorbidos, se añaden directamente al fichero de MUMs. De esta forma, aunque el usuario haga zoom en estos SMUMs, no se verán los MUMs que contienen. Aunque no es lo más realista (pues los SMUMs son approximate string matichings), es la mejor forma de limitar el tamaño del fichero de MUMs y a la vez mostrar las areas donde hay más similitud entre los dos genomas.

3.9 Automatización de todo el proceso

Todo el proceso para generar los archivos que se utilizan en la aplicación web consiste en: descargar genomas, parsear los archivos de genomas descargados, calcular las comparaciones entre genomas (obtener los MUMs), calcular el árbol filogenético, y preparar los archivos de MUMs y SMUMs para que se vean bien en la vista en detalle (MUMMY). Como este proceso se ha de repetir constantemente para mantener la aplicación web actualizada, se ha de automatizar para que los robots se ejecuten periódicamente, incorporando los nuevos genomas y recalculando los actualizados.

La automatización la he realizado con el programa cron de Linux. El programa cron permite ejecutar comandos cada cierto tiempo. Actualmente se lanzan una vez por semana los robots encargados de realizar todos los pasos del proceso. Si el lanzamiento de la semana anterior no ha finalizado, se espera a la semana siguiente terminando la ejecución actual (que no ha hecho más que comprobar si la anterior había terminado o no). Los lanzamientos del robot de eucariotas y los del robot de bacteria/arquea son completamente independientes, y no los lanzo simultáneamente. El lanzamiento para bacteria y archaea se lanza cada lunes, y el lanzamiento para eucariota cada jueves.

En el caso de bacterias y arqueas, ambos tipos de genoma utilizan el mismo robot. Como no pueden estar dos robots iguales ejecutándose al mismo tiempo, se ejecuta una semana el de bacterias y otra semana el de arqueas, alternándose. Esto lo he implementado escribiendo en un fichero un “0” cuando se termina una ejecución para genomas de bacteria del robot, y un “1” cuando se termina una ejecución para genomas de arquea. De esta forma, la próxima vez que el robot se ejecute (y ya haya acabado la ejecución anterior), si hay un “1” en el fichero, se lanzará una ejecución para genomas de bacteria, y si hay un “0” en el fichero, se lanzará una ejecución para genomas de arquea.

4 Resultados

Los robots de descarga de genomas de eucariota y genomas de bacteria/arquea funcionan correctamente y han sido adaptados a los cambios de formato de los archivos del servidor del NCBI. Los robots se encargan de descargar la lista de genomas actualizada del servidor y seleccionar qué genomas descargar. Se descargan todos los genomas nuevos que no están añadidos al árbol filogenético, y también todos los genomas que ya están en el árbol pero han sido actualizados después de la última descarga.

Todo el proceso que realiza el robot, desde descargar los genomas hasta la comparación de genomas, está automatizado para que se ejecute periódicamente. De esta forma se garantiza que siempre se tienen añadidos al árbol filogenético todos los genomas disponibles en el servidor del NCBI.

Actualmente, el árbol filogenético de genomas de eucariota cuenta con 31 genomas añadidos. Hay más de 160 genomas de eucariota en el servidor del NCBI, pero el proceso de comparación de genomas de ecuariotas es muy lento, así que aproximo que se tardará unos 6 meses en añadir todos los genomas que hay disponibles en este momento en el servidor del NCBI. En la figura 6 del apéndice se puede ver el árbol con los nombres de los genomas.

El servidor del NCBI cuenta con cuenta con 6004 genomas de bacterias, y los representantes de cada familia (2004 en total) de esas bacterias están ya añadidos al árbol filogenético de bacterias.

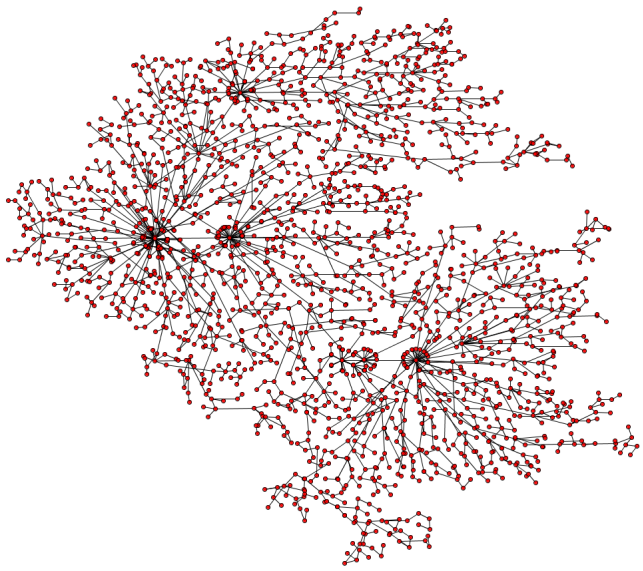

Fig. 3: Árbol filogenético de genomas de bacteria, 2004 genomas.

El árbol filogenético de genomas de arquea también está actualizado con todos los representantes de cada familia de arqueas que hay en el servidor del NCBI en este momento. Son 161 familias distintas actualmente.

La interfaz de vista en detalle de la comparación de genomas, llamada Mummy, también funciona correctamente con todas las comparaciones, tanto de eucariota como de barcteria y arquea. Se muestra primero una vista global de la similitud entre los genomas, mostrando los SMUMs (agrupaciones de MUMs). Cuando el usuario hace el suficiente zoom, se pasa a la vista de los MUMs, que son los exact matchings. En el caso de eucariotas se muestran diferentes niveles de SMUMs mostrando los más grandes en la visión global (big SMUMs), y mostrando SMUMs más pequeños en las zonas de menor similitud cuando te aproximas (SMUMs no absorbidos). Las líneas verdes representan el código genético compartido de forma directa (se encuentra en el mismo orden en los dos genomas). Las líneas rojas representan el código genético compartido de forma inversa (en uno de los dos genomas, la secuencia compartida está en un orden y en el otro genoma está al revés). Se pueden ver ambos tipos de similitud (directa e inversa) al mismo tiempo, o únicamente ver una de las dos.

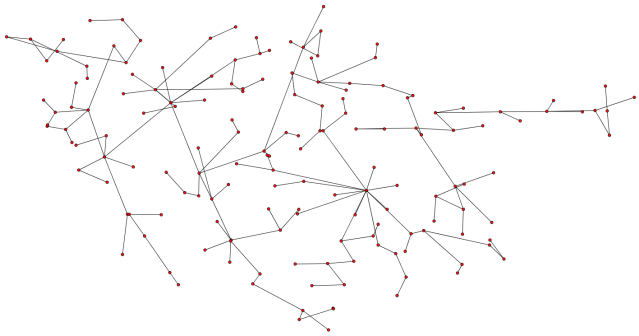

Fig. 4: Árbol filogenético de genomas de arquea, 161 genomas

En el caso de bacterias y arqueas, se pueden seleccionar tantos genomas como se quiera para comparar entre ellos. El Mummy solo puede mostrar 8 genomas por pestaña a la vez, pero se abrirán tantas pestañas como genomas hayan sido seleccionados. Si los archivos de MUMs y SMUMs de algunos de los genomas de bacterias o arqueas seleccionados no están previamente calculados, se calcularán en el momento. Si solo son 8 genomas de bacteria los que se comparan, o menos, se lanza su cálculo en paralelo. Si se abren múltiples ventanas, se lanzan de forma secuencial las comparaciones entre los genomas de cada ventana (para evitar la saturación del servidor si se abren demasiadas pestañas a la vez).

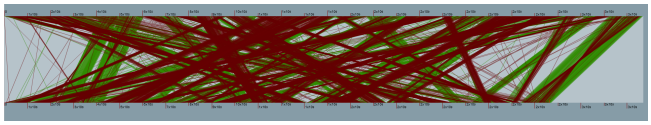

Fig. 5: Comparación de genomas de Callithrix_jacchus (primate) y Canis_lupus_familiaris (perro), zoom-out.

En el apéndice se pueden encontrar imágenes de la vista en detalle con mayor resolución y varios niveles de zoom.

5 Conclusiones

Gracias a este trabajo he tenido la posibilidad de afrontar un proyecto relacionado con el big data, con gran cantidad de datos, archivos de gran tamaño y largos tiempos de ejecución.

He aprendido que en el caso del big data es aún más importante no dejar ningún error sin corregir en un programa, por insignificante que parezca, ya que al realizarse tantas veces cada cálculo y con tantos casos distintos, si algo puede fallar, seguro que falla. También he aprendido lo importante que es la planificación del proyecto al trabajar con grandes cantidades de datos, puesto que los tiempos de cálculo son muy elevados y si la planificación no es correcta es muy fácil quedarse sin tiempo.

En el caso de este proyecto, la planificación fue hecha correctamente. El problema fueron las tareas que no se planearon en un principio. Gracias a que se dejó un margen de tiempo en la planificación para tareas imprevistas, no ha habido problema en realizar las tareas no planeadas.

Agradecimientos

Agradezco toda la ayuda prestada por el tutor del proyecto, Mario Huerta, ya que ha aportado muchas ideas sobre como afrontar los problemas que me iban surgiendo en el desarrollo del proyecto.

También agradezco el soporte que me han brindado mis padres y mis amigos durante la realización de este proyecto.

Referències

[1] http://ibb.uab.cat/ibb, Web del Instituto de Biotecnología i Biomedicina de la Universitat Autònoma de Barcelona.

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

[3] Suffix Tree Construction with slide nodes. Mario Huerta. Technical report LSI-02-63-R. Dept. Llenguatge i Sistemes Informàtics, Universitat Politècnica de Catalunya (2002).

[4] Efficient space and time multicomparison of genomes. Mario Huerta, Xavier Messeguer. Research Report LSI-02-64-R. Llenguatge i Sistemes Informàtics, Universitat Politècnica de Catalunya (2002).

[5] Identification of patterns in biological sequences at the ALGGEN server. PROMO and MALGEN. Domènec Farré, Romà Roset, Mario Huerta, José E. Adsuara, Llorenç Roselló, M. Mar Albà, Xavier Messeguer. Nucleic Acids Research 31(13): 3651-3653 (2003).

[6] http://platypus.uab.cat/, 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).

[7] http://www.ncbi.nlm.nih.gov/,WebServer del NCBI (National Center for Biotechnology Information).

APÉNDICE

A.1 Árbol filogenético de genomas de eucariota

Fig. 6: Árbol filogenético de genomas de eucariota, 31 genomas.

A.2 Vista en detalle de la comparación de genomas.

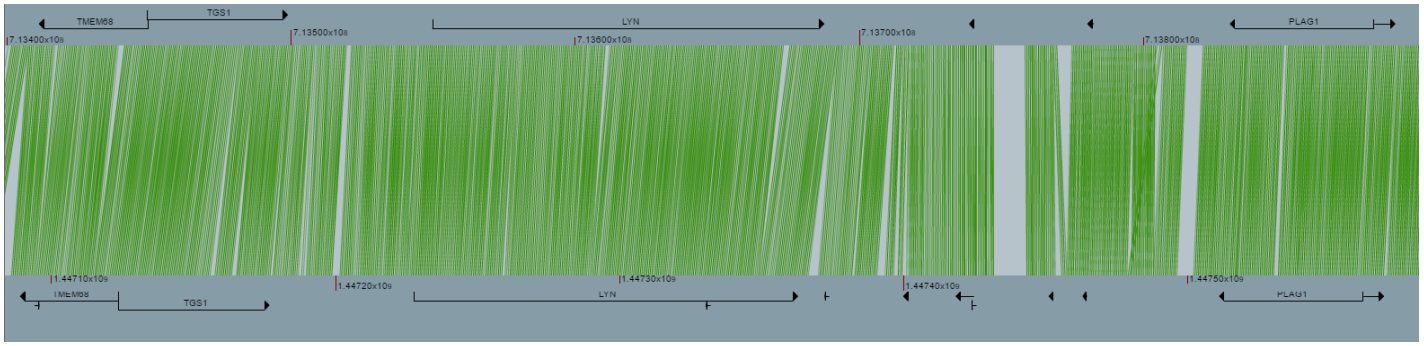

Fig. 7: Comparación de genomas de Chlorocebus_sabaeus (primate) y Homo_sapiens (humano), zona con genes compartidos, zoom-in.

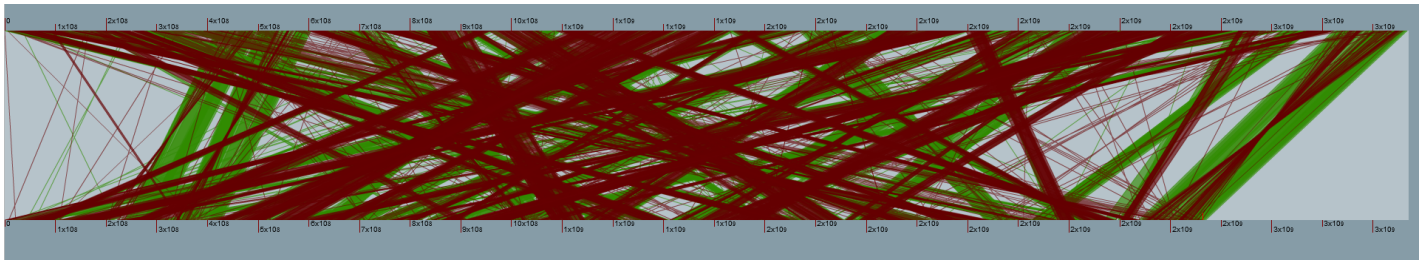

Fig. 8: Comparación de genomas de Callithrix_jacchus (primate) y Canis_lupus_familiaris (perro), zoom-out.

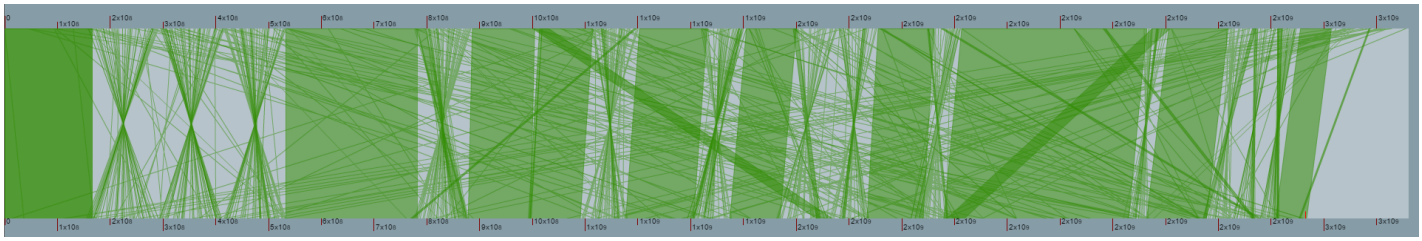

Fig. 9: Comparación de genomas de Bos_taurus (vaca) y Capra_hircus (cabra), zoom-out, solo directos.

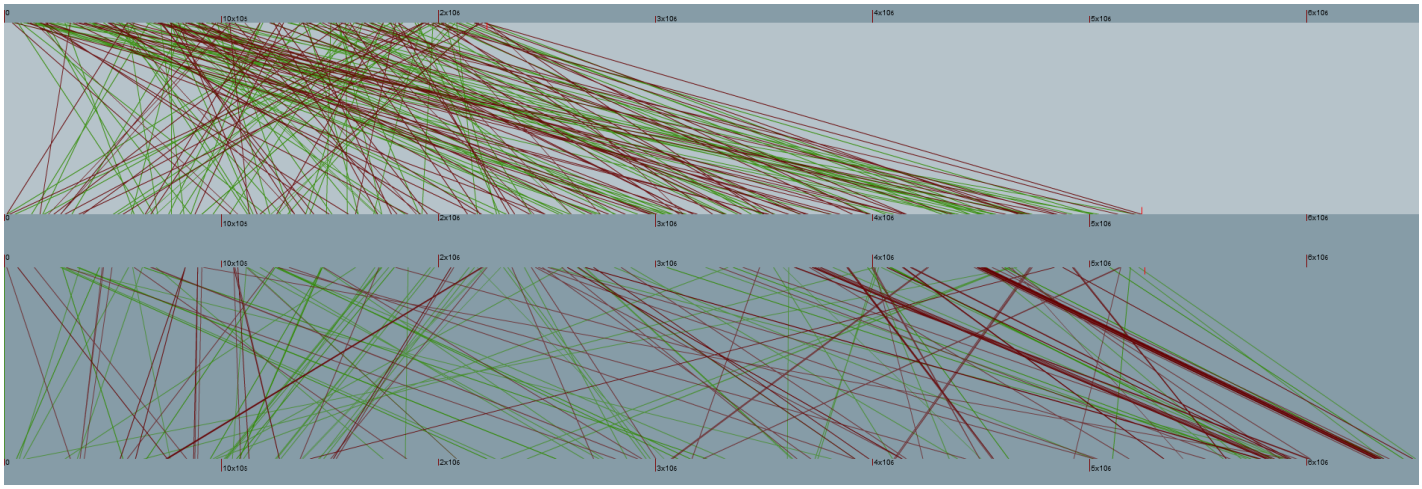

Fig. 10: Comparación de genomas de bacteria, tres genomas seleccionados.