Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Tesis de Máster

Máster en Ingeniería de Sistemas e Informática

Curso 2010/2011

Análisis filogenético molecular: Diseño e implementación de algoritmos escalables y fiables y verificación automática de propiedades de una filogenia.

Autor:

Jorge Álvarez Jarreta

Bajo la dirección de:

Elvira Mayordomo Cámara
Gregorio de Miguel Casado

Departamento de Informática e Ingeniería de Sistemas
Área de Lenguajes y Sistemas Informáticos
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Septiembre de 2011

Análisis filogenético molecular: Diseño e implementación de algoritmos escalables y fiables y verificación automática de propiedades de una filogenia.

RESUMEN

La filogenética es la ciencia que estudia las relaciones entre organismos basándose en lo cercano que están unos de otros. La forma más visual y conveniente de representar estas relaciones evolutivas entre un grupo de organismos es a través de los árboles filogenéticos. La secuenciación de cadenas biológicas es un proceso mecanizado que permite obtener las secuencias biológicas que se utilizarán para construir estos árboles, que se desarrollan a partir de modelos evolutivos: modelos matemáticos que intentan explicar de la forma más fiel posible la evolución real de dichas secuencias.

El análisis filogenético es un proceso formado por distintas etapas, que pueden variar según los objetivos, pero cuya finalidad es siempre la misma: poder reconstruir el árbol filogenético. Estas etapas pueden incluir: estudio de modelos evolutivos, análisis estadístico, alineamiento de secuencias, ...

Esta tesis de máster tiene como objetivo principal el desarrollo del criterio teórico y herramientas prácticas que permitan llevar a cabo un análisis filogenético completo.

Los procesos de secuenciación de cadenas biológicas no están exentos de errores, los cuales pueden aparecer en cualquier parte de la secuencia. Hasta la fecha, todo proceso de verificación de la secuencia requiere la actuación manual de un experto en el campo, lo que, sin duda, es un proceso muy costoso. Por otro lado, actualmente el coste computacional limita de forma práctica tanto la realización de filogenias extensivas (tratando miles y decenas de miles de secuencias) como la aplicación de modelos evolutivos más generales, interesantes y explicativos que el modelo uniforme (que únicamente se utilizan en tamaños de problema reducidos).

Las aportaciones de este trabajo abordan tres aspectos específicos: por un lado el desarrollo e implementación de un sistema de inferencia filogenética que concentra varios métodos sobre análisis de secuencias y estudio de filogenias que no se habían unido hasta el momento; por otro lado el desarrollo e implementación de una aplicación para la detección automática de errores de las cadenas obtenidas en los procesos de secuenciación; y por último el estudio teórico de nuevos algoritmos para la caracterización de problemas entre aquellos que se consideran como no resolubles en la actualidad.

Los resultados de todo este trabajo han concluido en la creación de dos artículos (uno publicado y otro en fase de revisión). Adicionalmente, hay un tercero que está en fase de desarrollo, lo que refleja el amplio interés en estos temas por la comunidad científica y su utilidad práctica.

Agradecimientos

En este rinconcito tan personal, dentro de un documento lleno de tanto formalismo y palabras técnicas, me gustaría destacar a unas cuantas personas que me han ayudado de forma especial a lo largo de este trabajo.

En primer lugar, como no podía ser de otra manera, me gustaría agradecer a Elvira toda su paciencia y dedicación que ha tenido conmigo durante este último año (el resto ya se lo agradecí en el PFC, tampoco quiero ser pelota), que incluso cuando sus proyectos personales han ocupado la mayor parte de su tiempo, ha conseguido sacar lo posible para ayudarme. En Goyo he descubierto un gran amigo, y me gustaría agradecerle el haberme aceptado en su vida privada cuando ya me sentía parte de la perspectiva laboral (y no tenía por qué extenderse). A Jose Manuel me gustaría agradecerle, sobre todo, esas charlas que hemos tenido, en las que he podido absorber algunos de esos conocimientos que solo la edad (¡que no vejez!) nos va dando. A Nacho y Roberto los meto en el mismo saco, ya que son mis camaradas, como Nacho suele decir, y me gustaría agradecerles las charlas, la ayuda y el apoyo que he recibido de ellos. No me gustaría olvidarme de Eduardo, que me ha hecho recordar más de una vez que cuando hablo de trabajo con gente de otros campos he de dejar de "hablar en chino" para que pueda existir una comunicación completa.

Alejándome un poco de la recién nacida Escuela de Ingenieros y Arquitectos, me gustaría dar las gracias a mis padres por su inestimable apoyo, cariño y, cómo no, sus broncas, que más de una vez me han venido bien para poner los pies en la tierra. A mis amigos Antonio y Fronti, quienes han tenido que aguantar más de una vez mis "rollos" bioinformáticos. Y, muy especialmente, a mi novia, Laura, por estar siempre ahí, y confesarle mi admiración secreta por su dedicación al trabajo, que me ha servido muchas veces de inspiración para querer superarme a mí mismo.

No quiero olvidar a quienes me han concedido una ayuda económica para poder continuar mi labor investigadora. Por ello, me gustaría agradecer al Instituto de Investigación e Ingeniería de Aragón (I3A), al Ministerio de Educación, al Departamento de Ciencia, Tecnología y Universidad del Gobierno de Aragón y al Fondo Social Europeo por ambas becas que he percibido durante este último año ([AP2008-03447] y [grupo GISED T27]).

Me gustaría terminar citando una frase de Anónimo, el autor inmortal e incesante de tantos escritos y dichos, "la prosperidad hace amistades, y la adversidad las prueba".

Índice general

Índice de figuras

Índice de tablas

1 Introducción

En este capítulo, como su título sugiere, se va a hacer una introducción de todos los aspectos que este trabajo aborda, incluyendo una breve descripción del contexto en el que se ha desarrollado y la estructura en la que se ha organizado el resto de la memoria.

1.1 Contexto del trabajo

La tesis de máster que se ha realizado prosigue, en uno de sus capítulos, el trabajo que desarrollé en el proyecto de fin de carrera el curso 2009/10 [3]. El resto de temas tratados en este trabajo pertenecen a nuevas vías de investigación exploradas a lo largo de este último año. Se ha desarrollado en el Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, dentro del ámbito de la bioinformática [28], en el grupo de investigación GISED.

En lo referente al contexto biológico, la tesis se centra en la rama de la filogenética, en otras palabras, en el estudio de la relación evolutiva entre organismos de la misma o distintas especies. En concreto, se ha trabajado con el ADN mitocondrial humano (mtDNA de ahora en adelante) como caso particular de estudio. Su análisis filogenético permite detectar enfermedades raras, muchas de ellas asociadas a una muerte prematura del individuo. La detección de estas enfermedades se basa en la localización de su ADN en el árbol de mtDNA [36]. Sin embargo, el espíritu de este trabajo se ha realizado tomando como premisa lograr la máxima generalización posible, permitiendo así abordar el estudio de información biológica de otro tipo de naturaleza (ADN nuclear, proteínas, etc.)

1.2 Objetivos

Una vez establecido el contexto de este trabajo, se concretan los siguientes objetivos de la investigación: la construcción de filogenias completas para grandes conjuntos de datos, la detección de errores de forma automática en secuencias biológicas y el estudio de nuevos algoritmos para problemas in-

tratables desde un punto de vista computacional. Estos temas, aunque aparentemente disjuntos, están muy relacionados, como se explicará brevemente a continuación y más en detalle en la introducción de cada capítulo.

El primero de los objetivos ha sido continuar el trabajo que se había realizado en mi proyecto fin de carrera con el objetivo de publicar un artículo, dado que la construcción de filogenias completas para grandes conjuntos de datos por sí es un tema muy interesante y poco abordado debido a su complejidad computacional. Además, con la adición del análisis de modelos evolutivos [42], un estudio estadístico exhaustivo y el uso de flujos de trabajo en su diseño [18], el entorno desarrollado presenta características novedosas en cuanto al problema objeto de estudio con resultados que confirman su viabilidad.

El segundo objetivo surgió con el análisis en profundidad del sistema desarrollado. El estudio estadístico resultó ser la parte donde se generaba un mayor número de tareas, y la justificación de esta etapa se basa en la incertidumbre existente respecto a la fiabilidad de las secuencias que se publican y almacenan en las bases de datos biológicas, como GenBank. Dado el crecimiento exponencial de estas bases de datos cada pocos meses [6], es inviable aplicar los procesos manuales de verificación de las secuencias que se realizaban hasta ahora. Por ello se ha visto la oportunidad de realizar un algoritmo de verificación automática de secuencias biológicas, que además tiene como valor añadido la supresión de la etapa del estudio estadístico del sistema de análisis filogenético, lo que mejorará considerablemente su tiempo de ejecución.

El tercer y último objetivo de este trabajo surge del interés de estudiar otros problemas para los que una solución algorítmica es intratable desde un punto de vista computacional, como sucedía en el caso del sistema de análisis de filogenias extensivas (con grandes cantidades de datos). Se han examinado varios casos que se han estudiado por otros autores centrados en la complejidad paramétrica [19], es decir, problemas para los que se puede dar solución algortímica fijando uno o varios de los parámetros que tienen como entrada. En concreto se ha realizado un estudio pormenorizado de un algoritmo concebido para la construcción de filogenias casi perfectas con el objetivo de incorporar criterios más sofisticados que se manejan en biología.

1.3 Estructura de la memoria

La memoria se ha dividido en seis capítulos. Los cinco que continúman tras esta introducción se describen brevemente a continuación.

El primer capítulo tras esta introducción tiene como título *Biología y bioinformática*. Reúne todos aquellos conceptos y definiciones, sobre todo de índole biológica, que se han considerado básicos y necesarios para poder comprender esta tesis en su totalidad.

El siguiente capítulo contempla toda la parte del trabajo relacionada con la construcción del sistema de inferencia filogenética mediante flujos de trabajo y selección de modelos evolutivos.

En el tercero se explicará todo el trabajo referente a la detección automática de errores en las secuencias biológicas obtenidas tras el proceso se secuenciación.

En el cuarto capítulo se desarrollan los aspectos esenciales relacionados con el estudio preliminar desde la perspectiva de la complejidad paramétrica del algoritmo para filogenias casi perfectas.

En el último capítulo se recogen todas las conclusiones relacionadas con los tres capítulos anteriores, junto con una valoración personal de conjunto del trabajo.

2
Biología y bioinformática

Este capítulo contiene todos aquellos términos y conceptos relacionados con la biología y la bioinformática que se han considerado fundamentales para la comprensión completa del trabajo. Tanto la estructura como el contenido han sido modificados y ampliados tomando como base el Capitulo 2 y el Apéndice B incluidos en mi proyecto fin de carrera [3].

2.1 Fundamentos biológicos

Las mitocondrias son unos orgánulos contenidos en algunas células en los que se produce la oxidación de las moléculas de glucosa, proceso por el cual la célula obtiene energía. El ADN mitocondrial humano (ADNmt de ahora en adelante) es un elemento biológico muy interesante desde muchos puntos de vista. Al estar dentro de las mitocondrias y ser independiente del ADN celular, es centro de muchas teorías y controversias respecto al origen o inclusión de estos corpúsculos en la célula, que es esencial para la vida de muchos seres vivos actualmente. El ADNmt posee tanto una tasa de mutación como de conservación muy elevadas, lo que lo hace idóneo para el estudio de individuos y su relación evolutiva, como miembros de una misma especie [20, 34, 43]. Además, el ADNmt está muy ligado a ciertas enfermedades que provocan una muerte prematura en los individuos que las padecen.

El ADNmt ha sido largamente estudiado, y actualmente se sabe que lo forman entre 16557 y 16576 nucleótidos, divididos en 37 genes distintos. Un gen es un segmento de una secuencia de ADN o ARN que contiene toda la información necesaria para codificar un elemento funcional de la célula. De esos 37 genes, 13 tienen como objetivo la creación de proteínas, 22 codifican ARNt (ARN transferente o de transferencia) y los dos restantes corresponden a las dos unidades que conforman el ribosoma (ARNr). Hay que indicar que el ADNmt es circular, a diferencia del ADN celular (muy conocido por su estructura de doble hélice). Posee una región de control, también conocida como bucle D, que no tiene como objetivo la codificación sino la unión de los extremos para conformar su estructura circular. Esta zona contiene dos regiones hipervariables: HVR_1 y HVR_2, las cuales pueden

determinar el haplogrupo de un organismo [37].

Un haplogrupo es una agrupación de haplotipos, los cuales representan conjuntos de polimorfismos con alguna característica estadística común. La clasificación y determinación de haplogrupos tiene como fundamento el detectar estas características comunes, las cuales pertenecerán de forma única a la familia en cuestión. Los haplogrupos son muy importantes en el estudio de la relación filogenética entre individuos, pudiendo determinar el punto de origen del linaje que se esté estudiando. En el caso del ser humano, el ancestro matrilineal común más reciente del ser humano se ha denominado "Eva mitocondrial" y vivió hace unos 140000 años. El ancestro común más reciente global, incluyendo la herencia patrilineal, es mucho más reciente.

Es necesario destacar la gran variabilidad del ADNmt, dado que es fundamental para poder realizar una clasificación en haplogrupos de forma fiable. Para dar cuenta de dicha fiabilidad, por ejemplo, se estima que entre dos seres humanos cualesquiera existen únicamente entre 50 y 70 nucleótidos diferentes, es decir, un 0.42 % del total de nucleótidos del ADNmt, una proporción muy elevada.

2.2 Introducción a la bioinformática

La bioinformática se puede considerar una rama de reciente aparición en la informática cuyo objetivo es el uso de la informática para la resolución de problemas biológicos. Su necesidad se ha visto acentuada en los últimos años debido a la magnitud y complejidad que están adquiriendo ciertas investigaciones referentes a la biología, que hacen intratable su resolución de forma manual [28, 24].

En concreto en este trabajo se han tratado temas del área de análisis de secuencias, más en concreto, sobre construcción de filogenias. Para poder desarrollar el estudio de filogenias es indispensable disponer de secuencias previamente alineadas y verificadas. El alineamiento de secuencias es una operación que establece equivalencias entre los distintos caracteres de las secuencias introducidas. Haciendo la suposición de que entre dichas secuencias existe un parentesco, pretende detectar aquellos hechos que las separan. Por otro lado, es importante que la información contenida en las secuencias haya sido verificada tras el proceso de secuenciación, dado que los errores introducidos en algunas posiciones podrían alterar la estructura final de la filogenia.

2.2.1. Filogenética

La filogenética la ciencia que tiene como objetivo la clasificación de las especies, tanto las existentes como las extinguidas, en base no a características fenotípicas sino a su relación evolutiva. Si el lector esta familiarizado con estos temas puede que vea en esta definición muchas coincidencias con

la cladística, y no se equivoca: a menudo filogenética y cladística son tratadas como sinónimos por tener el mismo objetivo.

La cladística tiende a definir distintos tipos de grupos de especies. Los más conocidos son:

Grupo monofilético o clado: compuesto por un organismo y todos sus descendientes. Al igual que pasaba con "Eva mitocondrial", la raíz u origen de este grupo es el ancestro común más reciente del mismo.

Grupo parafilético: compuesto por aquellos organismos cuya raíz u origen no incluye a todos los descendientes.

Grupo polifilético: compuesto por varios grupos monofiléticos no solapados. Dada su complejidad suele desaconsejarse su uso.

Estos grupos normalmente se representan en árboles biológicos donde se expresa la relación de parentesco entre organismos o especies (almacenados en sus hojas). Estos árboles se conocen como árboles filogenéticos. Aunque es común, sobre todo en la cladística, que estos árboles sean binarios, la aridad o número de subárboles que se derivan de cada nodo interno puede ser mayor que 2.

Modelos evolutivos

Los modelos evolutivos, también conocidos como modelos de substitución, son modelos matemáticos que se crearon para definir la probabilidad de cambio entre los distintos nucleótidos en secuencias de ADN o ARN, o aminoácidos, en el caso de proteínas. Su objetivo es intentar explicar la evolución que han sufrido dichas secuencias a lo largo del tiempo. El modelo puede contemplar la posibilidad de que exista reversión, es decir, que una mutación que se haya producido se deshaga con el paso del tiempo.

Un modelo queda completamente definido con la asignación, parcial o total, de una serie de parámetros. Los términos que quedan libres serán determinados en el momento que se evalúe el modelo, dependiendo su valor de las secuencias de entrada. Por ejemplo, algunos modelos establecen que el cambio entre nucleótidos en secuencias de ADN o ARN es equiprobable, mientras que otros postulan que las transiciones (cambio entre nucleótidos del mismo tipo) son más frecuentes que las transversiones (cambio entre nucleótidos de distinto tipo). Recordar que con tipos se hace referencia a la división entre purinas, que son los nucleótidos A y G, y las pirimidinas, C y T. No se debe olvidar que los modelos requieren que se asigne o calcule las frecuencias iniciales de los distintos nucleótidos o aminoácidos.

Selección de modelos

Actualmente existe una extensa variedad de modelos, y su elección, lejos de ser irrelevante, puede acarrear serios problemas en trabajos de análisis o

estudio de secuencias o proteínas. Esencialmente, una mala elección en un modelo evolutivo puede establecer relaciones incorrectas entre especímenes o, en el mejor de los casos, derivar en un árbol de peor calidad [27, 32, 42].

La complejidad y desconocimiento que se posee actualmente de la evolución tiene como consecuencia inmediata que ni el mejor de los árboles sea fiel a la realidad [40]. Por tanto, nunca se debe olvidar que la filogenética pretende obtener filogenias próximas a la realidad, pero en ningún momento se puede estar seguro de que la coincidencia sea total.

Aunque la importancia de la elección de un modelo evolutivo levanta opiniones contradictorias entre los investigadores, debido a los inconvenientes expuestos, se intenta demostrar la efectividad de estos modelos comparando los resultados que se obtienen con conjuntos de pruebas creados de forma artificial, de los cuales se conoce su relación evolutiva "real" y, por tanto, su modelo evolutivo correspondiente.

Existen varios métodos aplicables a un árbol filogenético para conseguir un valor que permita determinar qué modelos se aproximan más a la realidad intrínseca de los datos. Un método muy aplicado es el de máxima verosimilitud (Maximum Likelihood en inglés) [15]. Este método tiene como objetivo deducir los datos de entrada a partir de los resultados observables [40, 27, 41]. Asignando una probabilidad a las distintas mutaciones que ha podido sufrir una secuencia, se estima la distribución de probabilidad del espacio de árboles. Es uno de los métodos más flexibles pero, a su vez, resulta uno de los más costosos, computacionalmente hablando. Se sospecha que puede tratarse de un problema NP-completo; pero así como para otros método ha podido demostrarse, para el método de máxima verosimilitud sigue siendo únicamente una suposición.

Además, pese a presentar una aproximación más sencilla y pontente que otros métodos, la máxima verosimilitud tiene el inconveniente de valorar más positivamente a los modelos más complejos (aquellos con un mayor número de parámetros libres). Para compensar esta deficiencia se han desarrollado los criterios de información, cuya base se sustenta en penalizar aquellos modelos de mayor complejidad. Así, mediante una combinación de ambos parámetros (complejidad y verosimilitud), se puede escoger un modelo evolutivo equitativo, dejando que sea el investigador el que decida en último lugar. Como parte de los criterios de información se pueden encontrar el de Akaike (conocido como AIC) y el bayesiano (conocido como BIC).

2.2.2. Superárboles

Se obtiene como resultado de aunar varios árboles filogenéticos bajo una misma raíz. Normalmente se requiere que, para que el resultado tenga sentido, exista cierto solapamiento entre las hojas que componen los distintos árboles filogenéticos [38].

3

Flujos de trabajo con selección de modelos: una aproximación *multilocus* al análisis filogenético

En este capítulo se ha reflejado todo el trabajo de investigación desempeñado con el objetivo de desarrollar un sistema para el análisis filogenético completo con conjuntos de datos muy grandes (miles y decenas de miles de secuencias biológicas), también conocidas como filogenias extensivas. Como se ha indicado en el Capítulo 1, este trabajo ha partido de mi proyecto fin de carrera [3].

3.1 Introducción

A pesar de que la información de caracter biológico que se tiene actualmente es imperfecta, una gran cantidad de conocimientos científicos se ha ido acumulando a lo largo de las últimas décadas. Así, los avances técnicos en los computadores expanden continuamente las fronteras de lo que es viable computacionalmente. La inferencia de filogenias pertenece a una clase de problemas que se resisten a los enfoques de resolución convencionales debido a su naturaleza altamente combinatoria. Por otra parte, el uso de modelos evolutivos, que reflejan patrones de cambio específicos observados en distintos conjuntos de datos, es crucial para la obtención de filogenias lo más fieles posibles a la realidad. Sin embargo, este objetivo no ha sido abordado en aquellos casos en que los conjuntos de datos son grandes, debido a su elevado coste computacional asociado.

La paralelización es una técnica muy útil para reducir el tiempo de ejecución de los algoritmos (siempre que sea posible aplicarla), apoyándose en la gran cantidad de recursos computacionales que están disponibles, por ejemplo, en forma de procesadores independientes interconectados por grandes redes de comunicación. Los esfuerzos en esta dirección se han centrado, sobre todo, en la paralelización denominada "de grano fino" aplicada a los algoritmos estándar y el uso de técnicas algorítmicas para mejorar las implementaciones existentes [26, 39]. Algunas de estas empresas han obtenido

medidas excepcionalmente buenas, pese a que los algoritmos estándar no hayan sido diseñados teniendo en cuenta la concurrencia. Además, el número de tareas independientes que puede haber en un determinado momento es limitado, así como su carga individual, la cual podría poner trabas a esquemas de asignación simples o restringir su uso a redes de corta distancia o con baja latencia. Los esquemas tipo *maestro-esclavo* dominan esta clase de enfoques.

Por otra parte, los flujos de trabajo (*workflows*) son formalismos abstractos que describen tareas complejas formadas por subtareas relacionadas entre sí. Esta sistematización ha sido desarrollada de forma natural en entornos de empresa y manufactura. Cuando son utilizados en la experimentación científica, no se encuentran únicamente en la estructuración y documentación de experimentos, sino que además, si se dan las especificaciones e implementaciones adecuadas, forman la base de entornos de experimentación de ejecución automática [18]. Una gran cantidad de proyectos software se han desarrollado con esta finalidad, incluyendo muchos enfocados a las aplicaciones en bioinformática [25]. Desgraciadamente, los flujos de trabajo no están concebidos para expresar y manejar los posibles niveles de paralelización alcanzables, aunque, con esfuerzo y algunos cambios, se puede lograr una adaptación aproximada. Actualmente su elevada naturaleza interactiva es su mayor debilidad para alcanzar estos propósitos. El trabajo previo en el uso de flujos de trabajo en filogenética ha seguido esta misma filosofía [11].

Aunque los entornos de ejecución de bajo nivel y los entornos interactivos tengan distintas metas, ambos pueden beneficiarse consiguiendo costes computacionales inferiores. Para logarlo, bajo la propuesta de uso de flujos de trabajo que se va a exponer a lo largo de este capítulo, se deben incorporar los primeros dentro de los segundos, de forma anidada. Se defiende con la propuesta el uso eficiente de fuentes conocidas de tareas independientes y potencialmente concurrentes, y de información biológica conocida o inferida que simplifica el caso general en algoritmos no informados y ofrece soluciones de mayor calidad en un menor tiempo.

Con estas ideas en mente, el resto del capítulo aborda dos objetivos principales: en primer lugar, revelar los procesos concurrentes existentes a alto nivel en las aproximaciones tradicionales a los problemas de reconstrucción filogenética, incluyendo métodos de particionado que incrementan la granularidad y mejoran el rendimiento, que, adicionalmente, permiten incorporar criterios adicionales que se manejan en biología (por ejemplo, diferentes genes y *loci* evolucionan de forma diferente según muestran distintos estudios *multilocus*); y en segundo lugar, diseñar e implementar flujos de trabajo automáticos y arbitrariamente escalables, haciendo hincapié en lo relativo a la modularidad, la facilidad en el mantenimiento y la posibilidad de integrar nuevas etapas (como se ha hecho con la selección de modelos evolutivos).

3.2 Descomposición del problema

A continuación se va a centrar el estudio en el problema de la reconstrucción canónica en la filogenética computacional. Dado un conjunto de secuencias S alineadas, el objetivo es generar un árbol T que satisface (o se aproxima a) un cierto criterio de optimización. La conexión entre S y T se establece mediante la asignación de etiquetas a las hojas del árbol por las secuencias del conjunto inicial. Las dimensiones principales del problema, que gobiernan la complejidad del algoritmo, son el número de secuencias s, compartido por S y T, y la longitud de las secuencias l. Dicha longitud l designa el número total de elementos cladísticos del conjunto, y, equivalentemente, la longitud del alineamiento.

El problema de inferencia filogenética puede descomponerse en un conjunto de subproblemas independientes (como la selección de modelos o la robustez estadística,) para los que es posible encontrar en la literatura científica un abanico amplio de métodos algorítmicos. Esto hace posible incorporar mejoras parciales e incluirlas directamente en el flujo de trabajo de alto nivel. El interés de esta invesigación se centra en el método de evaluación estadística [21]. Esta inclusión viene impuesta por el deseo de poder evaluar ciertas características como la robustez y calidad de los resultados. La mayoría de estos métodos requieren que se establezca un número de réplicas estadísticas r que se pasará a un generador de muestras (probablemente junto a otros parámetros), el cual generará un número igual de alineamientos derivados, cada uno de los cuales constituirá un problema independiente de la misma magnitud al original, que será resuelto independientemente y agrupado posteriormente con el resto para poder obtener el resultado final. En este punto se encuentra el primer nivel de concurrencia originado por la independencia de los datos.

Además, se puede identificar en la selección de modelos [42] una tarea que la precede, y que además es apta para un tratamiento determinista. Mientras los parámetros a usar con el algoritmo de construcción del árbol son proporcionados por el usuario, es conveniente, de manera general, utilizar procesos de selección que evalúen un amplio rango de modelos M y que seleccionen aquél que se ajuste mejor al alineamiento de los datos. De nuevo, encontramos un flujo de trabajo muy sencillo compuesto por tantas tareas como modelos se tengan en consideración m; siendo cada uno independiente del resto. Al finalizar la evaluación de todos los modelos, se recopilan los resultados y se escoge aquel que haya obtenido mejor valoración. Esta tarea tiene lugar después del alineamiento y antes de que ninguna de los procesos asignados a cada muestra estadística comience su ejecución.

Esto en lo referente a concurrencia por independencia de los problemas. Sin embargo, se puede encontrar dentro de los propios datos una fuente de tareas independientes realizable por medios automatizados. Para empezar, los datos biológicos presentan estructura, y, de hecho, empiezan a aparecer

estudios multilocus cada vez más completos que tienen como objetivo explotar los beneficios derivados del uso de información extraída a priori [30, 29]. Esto quiere decir que es beneficioso hacer uso de la información que se puede extraer de los datos a priori. En este punto de vista, la preclasificación de acuerdo a hechos establecidos o hipótesis (a pesar de la prueba de las mismas por cualquier medio necesario) permite establecer particionados del conjunto de datos con dos principales beneficios: la generación de tareas independientes y el reducido tamaño de las mismas. A continuación se van a examinar los efectos de esta propuesta en las dos dimensiones fundamentales.

1. Considerese la naturaleza de los distintos caracteres cladísticos, representados por l. A pesar de la naturaleza homogénea de los alineamientos de las secuencias, el genoma está compuesto en realidad por zonas codificantes (genes) y no codificantes, a menudo afectadas de distinta forma e intensidad por la presión evolutiva. Por tanto, un alineamiento debería estar dividido en subconjuntos de columnas correspondientes a cada unidad genética, habiendo g en total. Así pues, basta con que el alineamiento tenga una secuencia con los g umbrales definidos para poder realizar la división de forma adecuada. Cada subalineamiento generado puede ser procesado como se ha descrito anteriormente (creando las muestras estadísticas y aplicando la selección de modelos), para combinar en una última etapa los resultados obtenidos con los de sus compañeros mediante algún método de coalescencia [13].

2. Mientras que las secuencias representan organismos individuales (s en términos de la complejidad del problema) y es tarea del análisis filogenético determinar su relación, sería posible identificar grupos de secuencias relacionados (evolutivamente hablando) de forma anticipada (haplogrupos en nuestro caso), y manejar estos grupos como unidades indivisibles de cara a la construcción del árbol. Un esquema de clasificación jerárquica sería apropiado para cumplir este objetivo, clasificando la pertenencia de cada secuencia a uno de los h grupos de subárboles, los cuales pueden ser construidos de forma independiente y congregados en la última etapa mediante algún algoritmo de generación de superárboles [7]. Se ha mostrado anteriormente cómo esta técnica ofrece grandes mejoras cuando se manejan grandes conjuntos de datos [9]. Obviamente, ambas estrategias pueden ser combinadas para conseguir un mayor efecto.

Nótese que el problema del alineamiento múltiple, el cual precede a todos los demás, puede beneficiarse de la aplicación de los mismos principios expuestos en los dos últimos párrafos. De hecho, ambas fuentes de información de la partición –una secuencia anotada para l y un clasificador de secuencias para s— son aplicadas habitualmente a las secuencias de forma individual, alineadas por pares con la secuencia de referencia.

3.3 Diseño del flujo de trabajo

El número y variedad de tareas independientes implicadas es realmente grande y ofrece muchas posibilidades de cara a una ejecución concurrente. La naturaleza de estas divisiones es a la vez simple y homogénea, formada por pasos de generación, de ejecución de múltiples instancias y de combinación, en los que la zona intermedia comprende la mayoría de la carga computacional. Los diferentes tipos de tareas están dispuestos de forma anidada: las operaciones de reducción generan grupos de datos relacionados, aunque independientes, hasta el momento en que los problemas más simples son resueltos y el proceso de clasificación se revierte mediante la combinación adecuada de algoritmos de agrupación (ver Figura 3.1).

Partiendo de estas consideraciones, se propone un flujo de trabajo modular basado en la definición de "cajas negras" reutilizables para cada una de las capas significativas de trabajo concurrente. La construcción presentada tiene un formato común, aunque único, y las variaciones que sea necesario incluir se pueden aplicar de forma sencilla. Cada una de las capas puede ser substituida por clasificadores triviales si se desea.

El problema básico de la computación de filogenias está formado por una etapa central y tres capas paralelas, como se explica a continuación.

Algoritmos de árboles. La etapa básica de la computación puede tener, de hecho, el mismo estilo que un flujo de trabajo, debido a la combinación de varios programas (los métodos de distancias son ejemplos muy comunes) o a las implementaciones de paralelismo a bajo nivel. Sea cual sea el caso, se puede suponer que esta "caja" toma el alineamiento A y un conjunto de parámetros y genera el árbol T.

Muestreo estadístico. La capa paralela fundamental comprende el muestreo estadístico y la solución concurrente a un número de problemas base, seguido de la aplicación de un algoritmo de consenso. La *interface* es semejante a la de la caja básica, exceptuando que es necesario indicar el número de réplicas r que determinan la magnitud total de la carga de trabajo asociada.

Árboles de genes. El preprocesamiento en l opera sobre alineamientos de secuencias sencillos y genera una serie de subalineamientos para ser posteriormente muestreados de acuerdo a las instrucciones de división representadas por S, las cuales son requeridas junto con los parámetros de sus tareas subordinadas. Es conveniente darse cuenta de que se puede suministrar un conjunto de modelos M, en vez de un único modelo μ, para elegir los parámetros más adecuados para cada gen, tal y como se explicará a continuación. Como en las anteriores cajas, su propósito es generar un único árbol T que dé explicación al alineamiento A.

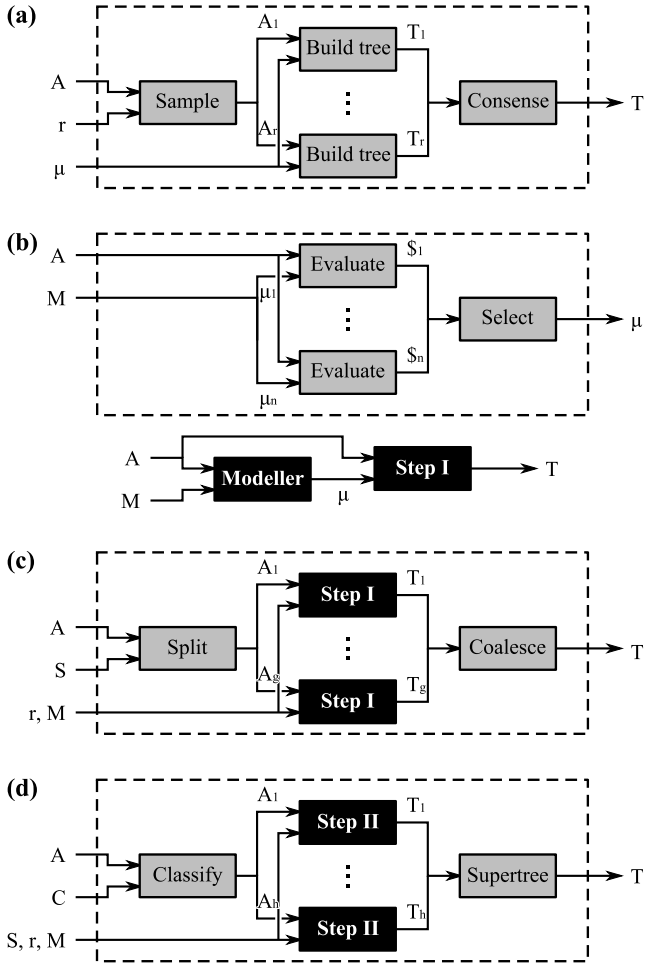

Figura 3.1: Jerarquía *bottom-up* de niveles concurrentes con el anidamiento y las interacciones con los problemas afines: **(a)** Nivel concurrente 1 (muestreo estadístico); **(b)** Selección concurrente del modelo e integración con el nivel 1; **(c)** Nivel concurrente 2 (árboles de genes); y **(d)** Nivel concurrente 3 (superárboles).

Superárboles. El tratamiento de s se consigue mediante un clasificador jerárquico C encargado de generar las entradas para cada subproblema, las cuales se pasan a la fase de árboles de genes explicada anteriormente. Cabe destacar que C podría ser incluido como entrada para el recolector de superárboles. El resto de sus parámetros pasan a la siguiente capa anidada, al igual que sucedía en las cajas anteriores.

Además, los siguientes problemas de precondición son también obvios objetivos para integrarlos en el flujo de trabajo.

Selección de modelos. Su objetivo, a diferencia de las capas anteriores,

es seleccionar un modelo μ de entre un conjunto de modelos M, de acuerdo con el alineamiento dado A, para lo que se debe indicar al programa de valoración algún método con el que poder medir lo ajustado que resulta cada modelo; a continuación, se evalúan los resultados (probablemente sesgándolos en contra de la complejidad de cada modelo) y se selecciona uno de los modelos. Este proceso está incorporado habitualmente al primer nivel, o, en algunos casos, como una etapa inmediatamente anterior.

Alineamiento de secuencias. Los niveles 2 y 3 pueden ser adaptados al problema del alineamiento de secuencias con un número de cambios mínimo. Operarán con conjuntos de secuencias no alineadas, en vez de con los alineamientos, donde cada clasificador dividirá esta colección de secuencias en otras más cortas o más pequeñas, según el caso. La estructura del flujo de trabajo permanece intacta, y solamente es necesario reemplazar los programas (o "cajas").

3.4 Aspectos de implementación

Cada tarea a realizar en una parte de la entrada (alineamiento parcial, análisis de modelos, reconstrucción de genes o subárboles) puede ser ejecutada independientemente del resto. El paralelismo resultante se ajusta perfectamente al uso de un clúster como entorno generalmente disponible y con una ejecución altamente flexible.

Dentro de todas las alternativas disponibles, se ha seleccionado Condor: un sistema de administración dedicado a la computación de alta productividad (o, como se conoce en su terminología anglosajona, *high-throughput computing*). Una de las herramientas que hacen de Condor una elección óptima es DAGMan (cuyas siglas se traducen del inglés como Gestor de Grafos Acíclicos Dirigidos) [12]: un *meta-scheduler* que permite diseñar una relación de orden entre procesos. El diseño del sistema no tiene ciclos, por lo que se puede realizar una traducción directa entre el diseño y un grafo acíclico dirigido (DAG en inglés). Es más, DAGMan ofrece la posibilidad de diseñar grafos anidados, es decir, DAGs en los que un nodo puede ser a su vez otro DAG, por lo que todos los nodos que dependan de este, deberán esperar a que el grafo anidado termine su ejecución correctamente para comenzar su tarea. Esta técnica, denominada en inglés *DAGMan within DAGMan*, contribuye de forma muy notoria a las propiedades de caja negra del sistema.

Los flujos de trabajo de Condor se generan automáticamente a partir de las entradas y sus clasificadores. Para una discusión detallada de las consideraciones técnicas, principalmente aquellas referentes al tamaño de los trabajos y a la gestión de procesos, ver [3].

Tabla 3.1: Resultados de la ejecución del sistema en el clúster con diferente número de muestras estadísticas.

Num. muestras	Num. tareas	CPUs total	CPUs usadas (media)	Coste secuencial (días)	Coste clúster (días)	*Speedup*
15	104782	400	200	291	3	97
200	287562	600	250	799	12	66.6

3.5 Resultados y análisis de rendimiento

En primer lugar se va a realizar una estimación de la complejidad del sistema. Como se ha descrito anteriormente, el sistema divide el alineamiento de entrada en h haplogrupos en primer lugar, y a continuación, cada haplogrupo es dividido en g genes. En el siguiente paso, m modelos son analizados para cada gen, seleccionando el mejor y finalmente, r muestreos estadísticos son generados y tratados con el modelo seleccionado. En resumen, el número total de trabajos involucrados en el sistema es $h \times (g \times (m + r + 3) + 2) + 2$.

Se ha probado el sistema con un alineamiento de 4895 secuencias completas de ADNmt humano real producido en el proyecto ZARAMIT para estudios filogenéticos exhaustivos [8]. Se han escogido $h = 26$ (el número de haplogrupos no vacíos en una clasificación básica), $g = 38$ (contando los 37 genes en el ADNmt humano y la región de control), y $m = 88$ (el conjunto de modelos incluidos en la versión 0.1.1 de la aplicación *jModelTest* [31], que además son los utilizados con mayor frecuencia en estudios sistemáticos). Substituyendo estos valores en la ecuación anterior obtenemos un total de $988 \times r + 89962$ tareas.

Se han realizado pruebas de viabilidad ($r = 15$) y pruebas a escala real ($r = 200$, dentro del rango $[100, 1000]$, un número de muestreos típico en estudios sistemáticos). La Tabla 3.1 resume los costes temporales real y secuencial (estimado) de las ejecuciones del sistema, así como otros datos relevantes. Señalar que para la estimación del coste secuencial del sistema se ha asumido que cada trabajo requiere 4 minutos de media para realizar su ejecución.

A continuación se va a proceder a estimar el tiempo de ejecución cuando se alcanza el máximo nivel de paralelización posible, en otras palabras, cuando todos los trabajos son ejecutados simultáneamente en distintos nodos del clúster. La partición más grande contiene el escenario del peor caso; en las pruebas realizadas, esto corresponde al gen MT-ND5 (1812 pares de bases alineadas) y al haplogrupo M (582 secuencias). Se han evaluado los 88 modelos en un ordenador con un procesador Intel Core 2 Duo y 8 GB de RAM para determinar cuál de los modelos es el más costoso, desde un punto de vista temporal, para el gen y haplogrupo mencionados. TVM+I+G

ha sido el peor modelo, con un coste temporal de 1 hora. Por lo tanto, el camino crítico del sistema tendría un coste de 2 horas y 20 minutos aproximadamente (incluyendo los trabajos intermedios y el coste de organización del propio clúster). Esto implica que en un clúster suficientemente grande, el sistema tardaría dicha cantidad de tiempo en realizar el estudio filogenético completo.

Para terminar la evaluación de los *speedups* iniciada en [9], se ha comparado la selección de modelo del sistema con el coste de la aplicación secuencial *jModelTest* (una de las más usadas). Para un subconjunto de 200 secuencias del alineamiento principal, *jModelTest* tardó algo más de 17 horas en evaluar los 88 modelos, mientras que el sistema necesitó poco más de 1 hora. En el futuro se presentarán algunos resultados con mayor nivel de detalle.

4

Detección automática de errores de secuenciación a partir de información filogenética

A continuación se va a exponer el trabajo concerniente al algoritmo creado para la detección de errores en secuencias biológicas tras el proceso de secuenciación. Esta idea surgió al observar el alto nivel de paralelismo que la fase de muestreo estadístico requería. Su necesidad en el análisis filogenético se sustenta en la inseguridad respecto a la fiabilidad de las secuencias biológicas, por lo que es necesario observar el cambio en la estructura de las filogenias al producir variaciones aleatorias en algunas posiciones de las secuencias a estudiar. Desarrollar una herramienta capaz de detectar los errores (para que sean corregidos) hace que esta fase ya no sea necesaria, mejorando de forma notoria el coste temporal del sistema explicado en el capítulo anterior.

4.1 Introducción

La continuación de los avances en las tecnologías de secuenciación de ADN desde 1970 ha permitido a la comunidad científica disponer de cantidades enormes de información biológica, cada vez a menor coste [23]. Desde que se obtuvo el primer borrador del genoma humano se han planteado nuevos objetivos más ambiciosos (y más asequibles) como la secuenciación del genoma completo y la medicina personalizada. Indudablemente, para que esto suceda, el problema de destapar los misterios del genoma debe ser completamente resuelto a todos los niveles. Sin embargo, esta situación ha conducido a un problema de saturación, dado que nos encontramos en un punto en el que la información es producida a mayor ritmo del que se puede asimilar, generando un desequilibrio que no deja de ampliarse a cada mejora que se introduce en la tecnología de secuenciación. La complejidad del genoma solamente puede ser comprendida explorando la variabilidad y efectos del mismo en las células y organismos, no a través de una única copia.

Pese a alguna pequeña bajada de la velocidad relativa, el crecimiento de

las bases de datos de secuencias públicas ha sido exponencial en los últimos 30 años, teniendo como referencia la duplicación actual del número de registros en GenBank cada 35 meses aproximadamente (encontrando algún caso de hasta 18 meses) [6]. En cualquier caso, no se espera una disminución en la velocidad de crecimiento del número de animales y genoma humano. Todo esto permite recoger conjuntos de datos genéticos razonablemente completos y representativos para grandes grupos de especies, teniendo disponible una cantidad elevada de información genética humana.

De todas formas, la situación no es tan negativa como parece en un principio, dado que la abundancia de material genético podría, de hecho, convertirse en una ventaja. La compatibilidad parcial en estudios *multilocus* se pueden consolidar a través de superárboles y métodos de coalescencia [7, 13], y se pueden formar filogenias robustas, al menos considerando un nivel medio-alto, incluso aunque necesiten ser sometidas a procesos de revaluación y refinamiento. Por lo tanto, es a los elementos que conforman las filogenias, es decir, las secuencias, a las que debemos dirigir nuestra atención.

Muchos de los contenidos de las bases de datos públicas no pueden someterse a curaciones independientes, en parte, debido al crecimiento que están sufriendo. Esto se traduce en que la mayoría de sus metadatos no están ni estandarizados ni son homogéneos. Por este motivo se puede afirmar, con cierta confianza, que la calidad individual de las secuencias de ADN, es decir, la precisión de la copia respecto a la original, es desconocida a priori. Los errores de secuenciación pueden ocurrir debido a la contaminación de la muestra –como fue el caso en una de las secuencias de referencia más conocidas, la del ADNmt [4]– como también a causa de las técnicas modernas de secuenciación de alta productividad, las cuales replican segmentos muy pequeños del ADN (al contrario de lo que hacían otros algoritmos más antiguos y menos eficientes) y después reconstruyen la secuencia mediante alineamientos locales de estos fragmentos, en los que los trozos más cortos son más susceptibles a generar falsos positivos. Los ratios de error de las tecnologías actuales son desconocidos, y la contaminación es claramente un factor no medible de forma aislada. Por el contrario, la disponibilidad de un gran número de secuencias muy estructuradas y mayormente correctas permite detectar y descartar los errores más claros, así como revelar características inusuales para ser confirmadas como novedades representativas o descartadas como elementos espurios.

El objetivo planteando en este capítulo es sistematizar la evaluación de nuevas secuencias de forma que ésta sea a la vez significativa e informativa. Para ello se van a utilizar colecciones de secuencias homogéneas como base para poder determinar la calidad de las nuevas secuencias, tomando como indicador el ajuste que éstas tengan a la estructura actual del conjunto. Este ajuste proporcionará también información suficiente como para poder señalar los errores potenciales para que sean revisados. Inicialmente, estos conjuntos pueden representar cualquier cosa, desde genes individuales hasta

genomas completos a través de un rango variable de organismos. Existen algunas comprobaciones básicas que pueden ser corroboradas directamente en los alineamientos de las secuencias, aunque sería muy beneficioso partir de una filogenia que ofrezca una clasificación natural y una jerarquía de las variaciones genéticas conocidas, por lo que se puede conseguir una verificación mucho más completa dentro de un contexto evolutivo.

En esencia, una filogenia provee un medio para agrupar cada nueva secuencia con sus parientes putativos y permite medir la similitud entre ésta y sus ancestros. Las mutaciones representativas de cada grupo tienen un alto índice de conservación, por lo que las excepciones a dichas mutaciones son ocasionadas en la mayoría de los casos por errores en el proceso de secuenciación (una vez se ha alcanzado una cierta profundidad en el árbol base). Es posible descubrir nuevos subgrupos y variaciones inusuales, y deberían ser tratados como excepciones legítimas, pero tan pronto como sean integradas en la filogenia (mejorando además la aproximación del árbol evolutivo) los descubrimientos extremadamente divergentes se irán volviendo cada vez más raros, hasta que desaparezcan. La interpretación de la alarma dependerá del estado actual de la filogenia guía y de las secuencias en consideración.

Un hecho interesante es que el mismo proceso seguido para determinar lo buena que es una secuencia está conectado al proceso de construcción de filogenias, pues se analiza dicha secuencia según el lugar más adecuado en el que se sitúa dentro de la filogenia. Esta es una característica adicional muy interesante, pues se ha mostrado que aunque las actualizaciones periódicas de grandes filogenias puede ser viable a través de métodos estándar [10], todavía resulta un proceso muy costoso cuya frecuencia para actualizar la filogenia debe ser limitada, lo que se deriva en una degradación y empobrecimiento de la misma en ese lapso de tiempo. Por otro lado, un árbol bien fundamentado y con una estructura principal estable a largo plazo, facilita el proceso de localización de la nueva secuencia, además de que su incorporación al árbol, si no exacta, afectaría, como mucho, de forma localizada. Sin duda, es difícil lograr un máximo beneficio en el compromiso entre estas dos perspectivas.

En resumen, en el resto del capítulo se va a presentar un algoritmo inspirado en la filogenética que tiene como objetivo evaluar la precisión de las secuencias biológicas a través de la historia evolutiva incorporada en el árbol de referencia. Estableciendo el valor de los umbrales de forma adecuada, el algoritmo será capaz de determinar cuándo una secuencia es correcta (sus mutaciones características no entran en contradicción con la estructura evolutiva manifestada en la filogenia), cuándo puede ser potencialmente correcta aunque inusual y cuándo ésta se desvía demasiado de la filogenia, por lo que se requerirá de una revisión cuidadosa.

4.2 Detección de errores de secuenciación

Para poder determinar si una mutación es real o puede haber sido generada en el proceso de secuenciación, el algoritmo requiere algún elemento con el que poder comparar la secuencia. Un árbol filogenético contiene una gran cantidad de información referente a la evolución a lo largo del tiempo de un tipo de estructura biológica determinada (ADN nuclear, ADNmt, proteinas, ...), por lo que es un candidato idóneo con el que realizar la comparación de la nueva secuencia. Como es obvio, es necesario que el árbol seleccionado haya sido construido bajo un modelo evolutivo conocido y aceptado, y con secuencias verificadas minuciosamente, de forma que no se hayan introducido errores en la fase de construcción de la filogenia. Esta última hipótesis es muy importante debido a que, si una rama queda definida por ciertas mutaciones erróneas, se podrían validar secuencias con errores en esas mismas posiciones.

Como se ha mencionado anteriormente, el proceso principal del algoritmo presentado se basa en la localización del lugar en el árbol filogenético donde la secuencia de entrada se ajusta mejor. Antes de exponer su funcionamiento de forma más detallada, se van a explicar dos operaciones básicas que serán necesarias a lo largo de la ejecución del algoritmo.

La primera es el *Filtro Hamming*. Esta operación toma como entrada dos secuencias de la misma longitud y provee como salida su distancia Hamming, es decir, el número total de posiciones en las que las dos secuencias no comparten el mismo valor (incluyendo en las posibilidades tanto los huecos como el carácter N o "desconocido").

La segunda es el *Filtro de Referencia*. Antes de ejecutar el algoritmo, se debe seleccionar una secuencia de referencia (que tendrá que estar incluida en el árbol filogenético). Como entrada, la operación toma dos secuencias, de las cuales una será la que se da como secuencia a analizar por el algoritmo y la otra pertenecerá al árbol. Primero, la operación obtiene una lista de las posiciones donde la secuencia de referencia difiere de la secuencia del árbol. Dado el hecho de que los huecos introducidos en la secuencia de referencia son debidos al alineamiento previo para la construcción de la filogenia, estas posiciones serán ignoradas. A continuación, se comparan los valores de la secuencia del árbol y de la secuencia de entrada en las posiciones incluidas en la lista obtenida en el paso anterior. Como salida se obtiene el número de diferencias resultantes de esta comparación.

El primer paso del algoritmo es alinear la secuencia de entrada con las secuencias del árbol. Esto significa que las secuencias del árbol filogenético deben estar previamente alineadas. Para este paso se ha utilizado MUSCLE [14], una herramienta para procesos de multialineamiento, añadiendo la nueva secuencia al alineamiento del árbol.

Después, el algoritmo localiza el nodo más próximo a la secuencia nueva. Para ello toma un nodo, al que denominaremos *padre*, y todos los nodos a

un nivel de distancia por debajo, sus nodos *hijo*. A continuación aplica el *Filtro de Referencia* a los pares generados con la secuencia de entrada y cada una de las secuencias de los nodos seleccionados. Normalmente, uno de los nodos *hijo* es seleccionado como el más cercano de todos los pares evaluados, por lo que se establece este nodo como un nuevo nodo *padre* y se repite el proceso hasta que el algoritmo alcance a un nodo hoja. Como es obvio, el primer nodo seleccionado es la raíz del árbol.

Existen otros casos que podrían darse en vez del caso común presentado. En vez de un único nodo, se pueden obtener dos o más como nodos más cercanos a la secuencia. Si todos los nodos son nodos *hijo*, el algoritmo explora cada una de las posibilidades independientemente, aplicando el proceso de forma individual. Las pruebas mostradas en la siguiente sección demuestran que esta situación con múltiples caminos no se mantiene normalmente más allá de dos o tres iteraciones. Si uno de los nodos cercanos es el nodo *padre*, éste es descartado en la medida en que preferimos un resultado mas cercano a las hojas. Las pruebas han revelado algunas situaciones a las que se han denominado *mínimos locales*, donde el *padre* resulta ser el único nodo más cercano, pero, como su nombre indica, se trata de una situación local: existen otros nodos, más próximos a las hojas, que son más cercanos a la secuencia que el nodo *padre*. Para poder eludir estos *mínimos locales*, el algoritmo aplica el *Filtro Hamming* a los mismos pares manejados en el *Filtro de Referencia*. Los diferentes resultados son procesados como en los casos expuestos anteriormente, exceptuando el caso en el que se obtenga de nuevo el *padre* como único nodo cercano. En este caso, se ha alcanzado un *mínimo global*, por lo que ese nodo del árbol es el más cercano a la secuencia nueva.

En un último paso, el algoritmo aplica nuevamente el *Filtro de Referencia* al nodo seleccionado para obtener el número total de diferencias con la nueva secuencia. Dos umbrales determinarán si la secuencia es correcta (*Right*), si existen posibles errores (*Alarm*), o si es, con bastante seguridad, errónea (*Wrong*). Es importante saber que estos umbrales no funcionarán de forma adecuada si la secuencia de entrada corresponde a alguna especie desconocida para la filogenia, u otros casos similares, que se reflejan como "agujeros" en el árbol filogenético.

Intuitivamente, debido a las situaciones de múltiples caminos, el algoritmo podría mostrar más de un nodo solución. Mirando de forma detenida al árbol se ha observado que todas estas soluciones con parientes cercanos entre sí, es decir, nodos con el mismo nodo padre o sobrino.

4.3 Pruebas y resultados

Como se ha destacado anteriormente, la detección de variantes inusuales requiere de una filogenia relativamente estable y muy poblada. Por el momento hay pocas instancias donde la cobertura sea suficientemente elevada

como para asegurar robustez. Posiblemente el más significativo de todos es el ADNmt humano, dado que es fácil de secuenciar, no recombinante (por lo que puede ser utilizado en bloque en la reconstrucción filogenética) y altamente informativo [5]. La mayoría de las áreas de la filogenia mitocondrial humana están fielmente representadas y las mutaciones características por las cuales grandes grupos de individuos son relacionados están organizadas en jerarquías extensivas de haplogrupos mitocondriales [44].

De hecho, el algoritmo toma la inspiración de los procedimientos aplicados en la construcción incremental de la filogenia de MITOMAP [35]. Mientras se combinan estos procesos con reconstrucciones estrictas, es posible aplicar los resultados del algoritmo para construir copias de la filogenia mitocondrial que reflejen las últimas incorporaciones al conjunto de secuencias de ADNmt humano publicadas. Esto es de gran utilidad debido a que la mayor parte de las secuencias pueden ser situadas con gran precisión en el árbol, lo que tiene el valor añadido de tener filogenias permanentemente actualizadas y siempre disponibles.

En referencia, las actualizaciones actuales a la filogenia mitocondrial humana ZARAMIT [8] requieren por encima de un año de tiempo de CPU secuencial (tiempo que puede ser reducido a semanas mediante el uso apropiado de procesamiento paralelo). Es claramente inviable y muy ineficiente producir actualizaciones cada vez que unas pocas secuencias son publicadas, pues su valor incremental no justifica el gasto adicional de recursos computacionales. No obstante, el número de adiciones entre reconstrucciones, por ejemplo cada pocos meses, puede ser extremadamente significativo. Por ejemplo, durante los primeros seis meses del 2011 aproximadamente 1000 secuencias nuevas fueron publicadas en GenBank, lo que implica un crecimiento del 14 % de la colección de secuencias mitocondriales completas acumuladas principalmente a lo largo de la última década.

Para los experimentos realizados se ha utilizado el último árbol filogenético creado por el proyecto ZARAMIT, compuesto por 7390 secuencias de ADNmt humano obtenidas de GenBank. Como secuencia de referencia, se ha utilizado la secuencia de referencia revisada de Cambridge (rCRS). Los umbrales para el *Filtro de Referencia* se han establecido en 0 para discriminar entre los estados *Right* y *Alarm*, y 3 para la distinción entre *Alarm* y *Wrong*.

4.3.1. Estudio de comportamiento

Para poder estudiar el comportamiento del algoritmo, se han dividido los experimentos en tres grupos, cada uno centrado en obtener unos resultados específicos dentro de todos los casos posibles.

1. **Localización correcta de las hojas:** El primer experimento tiene como objetivo localizar correctamente algunas de las secuencias que for-

Tabla 4.1: Secuencias pertenecientes al árbol filogenético y su clasificación por el algoritmo.

Accession	Ref.	Clasificación	Localización	Distancia
DQ246811	[33]	RIGHT	DQ246811	0
DQ246826	[33]	RIGHT	DQ246826	0
DQ246828	[33]	RIGHT	DQ246828	0
DQ246830	[33]	RIGHT	Anc3521, Anc3534	271
AY738944	[1]	RIGHT	AY738944	0
AY738945	[1]	RIGHT	AY738945	0
AY738946	[1]	RIGHT	AY738946	0
AY738947	[1]	RIGHT	AY738947	0
AY738948	[1]	RIGHT	AY738948	0
AY738949	[1]	RIGHT	AY738949	0
AY738957	[1]	RIGHT	AY738957	0
AY738958	[1]	ALARM(1)	Anc4104, Anc3956	7
AY738980	[1]	RIGHT	AY738980	0
AY738981	[1]	RIGHT	Anc4051, ...	2
AY738982	[1]	RIGHT	AY738982	0
AY738990	[1]	RIGHT	AY738990	0
AY738991	[1]	RIGHT	AY738991	0
AY738992	[1]	RIGHT	AY738992	0
AY738993	[1]	RIGHT	AY738993	0
AY738994	[1]	RIGHT	AY738994	0

man parte de las hojas del árbol filogenético. Específicamente, se han seleccionado 20 secuencias del conjunto de hojas del árbol. Los *accession* así como los resultados obtenidos por el algoritmo se presentan en la Tabla 4.1.

Como se puede ver en la Tabla 4.1, 17 secuencias se han localizado correctamente. A pesar de que la clasificación de dos de las tres restantes ha sido *Right*, el algoritmo no ha sido capaz de localizarlas en el árbol. La primera, DQ246830, tiene una distancia enorme con los nodos más cercanos, hecho que no sucede con la segunda, AY738981. Si observamos la primera de estas secuencias, se puede ver que el primer fragmento de la región de control no está, por lo que es normal haber

Tabla 4.2: Secuencias de distintos animales y su clasificación por el algoritmo.

Accession	Animal	Clasificación	Distancia
NC_001643	Chimpancé	RIGHT	1966
NC_001941	Oveja	RIGHT	4720
NC_007402	Serpiente de rayo de sol	RIGHT	6324
NC_006160	Mosca blanca	RIGHT	9303
NC_005313	Atún bala	WRONG(7)	5779
NC_009885	Nematodo	RIGHT	9077
NC_006281	Cangrejo azul	RIGHT	9251
NC_005805	Rana de árbol moteada	RIGHT	6191
NC_008159	Coral de setas	WRONG(6)	8239
NC_009684	Pato silvestre	WRONG(6)	5822

obtenido esa distancia y que el algoritmo no haya sido capaz de localizar la secuencia. En el segundo caso, el algoritmo no ha podido localizar la secuencia, pero en los resultados está incluido el nodo Anc4051 y otros nodos y hojas que son hijos del mismo nodo. El Anc4051 es el bisabuelo de AY738981, por lo que el resultado es muy próximo a la solución ideal. Esto puede suceder con algunas secuencias debido a la dependencia existente entre el algoritmo y el árbol. Algo parecido sucede con la secuencia AY738958, solo que en este caso una de las mutaciones no se ajusta exactamente al clúster donde ha sido situada, por lo que se ha clasificado como *Alarm*.

2. ADN mitocondrial no humano: En estos experimentos se han utilizado secuencias procedentes de otros animales, de forma que se pueda comprobar el comportamiento del algoritmo con secuencias que no se ajustan a un árbol filogenético de ADNmt. Los animales específicos y los *accessions* de las secuencias se muestran en la Tabla 4.2.

El primer elemento que llama la atención de los resultados es que la mayoría de las secuencias se han clasificado como *Right*. Este hecho está relacionado con el prealineamiento realizado por el algoritmo con la secuencia rCRS, lo que puede alterar el valor de algunas posiciones para obtener el mejor resultado posible. Dada esta situación, el resultado más importante de estos experimentos es el campo *Distancia*. No es sorprendente que la secuencia del chimpancé sea la más cercana de entre todas las secuencias examinadas. En la mayoría de los casos, la distancia implica que más del 30 % de los nucleótidos están mal, lo que es otra señal inequívoca de que la secuencia no se ajusta al árbol

Tabla 4.3: Secuencias sintéticas creadas a partir de la secuencia AY738958.

Accession	Mutación
SEQ00001	-3106A
SEQ00002	-3106A, G8859A
SEQ00003	-3106A, G8859A, G15325A
SEQ00004	T6775C
SEQ00005	T6775C, G1437A

Tabla 4.4: Clasificación de las secuencias sintéticas por el algoritmo.

Accession	Clasificación	Localización	Distancia
AY738958	ALARM(1)	Anc4104, Anc3956	7
SEQ00001	ALARM(2)	Anc4104, Anc3956	8
SEQ00002	ALARM(3)	Anc4104, Anc3956	9
SEQ00003	WRONG(4)	Anc4104, Anc3956	10
SEQ00004	RIGHT	Anc4076	7
SEQ00005	ALARM(1)	Anc4076	8

y sería aconsejable comprobar si pertenece a un *Homo sapiens*.

3. Mutaciones sintéticas: Con estos experimentos se pretende probar que el algoritmo es capaz de detectar cada mutación relevante de forma individual. Se ha tomado la secuencia AY738958 como secuencia base a la que se han introducido de forma artificial algunas mutaciones para observar cómo cambia el resultado del algoritmo. Estas mutaciones se muestran en la Tabla 4.3.

Como suele ser el formato habitual en biología, las mutaciones se codifican mostrando el valor anterior, la posición y el nuevo valor asignado a dicha posición. La Tabla 4.4 contiene los resultados de cada una de las secuencias analizadas, mostrando de nuevo el resultado de la secuencia AY738958 para poder ver cómo las mutaciones introducidas han afectado a los resultados.

Las primeras tres secuencias sintéticas demuestran cómo una sola mutación, aplicada en el sitio adecuado, puede cambiar una clasificación de *Alarm* a *Wrong*. Las últimas dos reflejan cómo, obviamente, una mutación puede cambiar el nodo más cercano. Normalmente, como en este caso, el resultado cambiará de un nodo a otro nodo hermano, por lo que no será un cambio demasiado relevante. Pero si tres o cuatro

mutaciones o errores se acumulan a lo largo de la secuencia en las posiciones adecuadas, se pueden obtener como nodos cercanos unos verdaderamente alejados de la localización real que tendría la secuencia en el árbol filogenético.

4.3.2. Estudio de rendimiento

Todos los experimentos se han ejecutado en un ordenador con un procesador Core 2 Duo E6750 y 8 GB de RAM. El coste temporal que implica cargar el árbol filogenético y toda la información necesaria por la aplicación supone, como máximo, 20 segundos, teniendo en cuenta que estos datos solo deberán ser cargados la primera vez, cuando se inicie la aplicación. El programa tarda 20 segundos de media en alinear y localizar la secuencia de entrada. Por tanto, el programa tiene un rendimiento excelente y el usuario puede obtener los resultados en "tiempo real", generando además una realimentación hacia el usuario al mostrar las posiciones que se han marcado como malas (si las hay) con respecto a los nodos más cercanos. El peor caso de los experimentos realizados se ha dado al utilizar como entrada la secuencia de la serpiente de rayo de sol, donde al programa ha tardado 32 segundos alinearla y localizarla.

5

Complejidad paramétrica en bioinformática: *filogenias casi perfectas*

En este capítulo se va a mostrar la investigación actualmente en curso sobre complejidad paramétrica aplicada a la bioinformática. En concreto, se centra el caso de algoritmos teóricos que tienen como objetivo construir filogenias perfectas y casi perfectas.

5.1 Introducción

Los problemas clasificados como NP-duros son realmente difíciles de resolver de forma eficiente y óptima debido, principalmente, a su alta explosión combinatoria. Es común encontrar algoritmos con heurísticas o aproximaciones para este tipo de problemas, que obtienen la solución con un coste computacional aceptable, aunque no aseguran hallar la solución óptima.

Un subconjunto de estos problemas NP-duros tiene una característica especial: su complejidad depende de la *estructura* de la entrada. Este tipo de casos permiten una descomposición del problema en un conjunto de parámetros, los cuales afectan de forma distinta a la complejidad del problema. El Tratamiento por Fijación de Parámetros, en inglés, Fixed-Parameter Tractablility (FPT), es una técnica aplicada a este tipo de casos que permite estudiar subconjuntos del problema fijando aquellos parámetros que afecten de forma más severa a su complejidad. De esta forma, se pueden llegar a desarrollar algoritmos eficientes para subproblemas, aún interesantes, del problema inicial, que además obtengan la solución óptima. Pese a ser una herramienta puramente teórica, su uso está muy extendido. Como ejemplo, el FPT ha sido utilizado en problemas de teoría de grafos, como el cubrimiento de vértices, entre otros [22].

Como ya se ha comentado en los capítulos anteriores, en bioinformática existen muchos problemas que aún no se han podido resolver debido a su coste computacional. El enfoque FPT constituye, en estos casos, una herramienta muy útil para poder obtener un algoritmo que resuelva aquellos subproblemas "sencillos" pero igualmente interesantes. Dentro de la filogenética, se ha utilizado tanto para problemas relacionados con su construcción, como

para la evaluación de modelos evolutivos, o la medición de distancia entre filogenias, entre otros [19].

En concreto, el trabajo que se va a mostrar en este capítulo se ha centrado en los problemas de construcción de filogenias, más específicamente, filogenias perfectas y casi perfectas. El problema de construcción de filogenias perfectas ha sido ampliamente estudiado [16], ypara el cual ya se han desarrollado algoritmos haciendo uso del FPT [2]. El objetivo de este problema es conseguir construir una filogenia que satisfaga las siguientes condiciones:

1. Entre dos secuencias unidas por una arista del árbol, solo debe existir, como mucho, un cambio de valor (mutación) en una de las posiciones de las secuencias.

2. El subgrafo del árbol inducido por un estado concreto en una posición de todas las secuencias debe ser conexo.

En concreto, el problema de encontrar una filogenia perfecta se engloba dentro del conjunto de problemas de árboles de Steiner. En este contexto, es posible encontrar en la literatura científica un algoritmo que da solución al problema de construcción de filogenias casi perfectas [17]. En este caso, el problema plantea si, dado un conjunto de secuencias S, existe la posibilidad de construir una filogenia con una penalización, como máximo, q. La penalización es un valor que representa el conjunto de ramas del árbol que no satisfacen las características de filogenia perfecta presentada anteriormente. Obviamente, es necesario establecer un criterio para generar este valor y determinar si la filogenia es solución al problema o no (y si es óptima). En la solución que se propone en [17] se ha utilizado un criterio de máxima parsimonia (MP), el cual simplemente suma la distancia Hamming (número de posiciones con valor distinto) entre todo par de secuencias unidas por una rama en el árbol, y luego resta el valor que se obtendría del mismo árbol en caso de tratarse de una filogenia perfecta.

La MP es un criterio muy sencillo, el cuál ha sido muy criticado por los biólogos debido a que no hace uso de ningún tipo de criterio biológico (como los modelos evolutivos en la máxima verosimilitud). Por tanto, el objetivo que se presenta en este capítulo y en el que se está trabajando actualmente, es realizar los cambios necesarios en el algoritmo de construcción de filogenias casi perfectas para que, en vez de utilizar el criterio de MP, utilice el criterio de máxima verosimilitud (MV).

5.2 Preliminares

En esta sección se van a presentar, un primer lugar, la notación y conceptos básicos para comprender el resto del trabajo que se presenta, y a continuación se muestran los dos algoritmos que aparecen en [17].

5.2.1. Notación

Dado un conjunto de secuencias S, $n = |S|$ indica el número de secuencias del conjunto. Del mismo modo, siendo C el conjunto de columnas de una secuencia, $m = |C|$ representa la longitud de la secuencia. Para una columna $c \epsilon C$, A_c es el conjunto de valores o estados que puede tener una secuencia en esa columna c. $r_c = |A_c|$ y, además, $r = max_{c \epsilon C} r_c$.

Sea T un árbol y $\sigma \epsilon A_c$. $T[\sigma]$ representa el subgrafo inducido por todos los nodos v tal que $c(v) = \sigma$. Sea $C_p \subseteq C$, C_p representa el conjunto de columnas c que se requiere que sean convexas, es decir, que $\forall \sigma \epsilon A_c$, $T[\sigma]$ es un grafo conexo. Si T es una filogenia y $\forall c \epsilon C_p$, c es convexa en T, entonces T es una filogenia C_p-perfecta. Cuando una filogenia es perfecta, se cumple que $C_p = C$.

Sea $Q \subset S$. Q es una subfamilia de caracteres si ninguna secuencia de Q comparte un estado con cualquier secuencia de $S - Q$ para algún c. Además, Q será adecuada si $\forall c \epsilon C_p$ Q y S-Q comparten como mucho un estado. A partir de ahora, se asume que siempre que se hable de subfamilia de caracteres se referirá a una subfamilia de caracteres adecuada.

El vector de corte de una subfamilia de caracteres Q es la secuencia $Sv(Q)$ donde, para cada posición c, si $c \epsilon C_p$ y Q y $S - Q$ comparten un valor σ en c, $c(Sv(Q)) = \sigma$; sino $c(Sv(Q)) = *$.

La función α asigna el estado de penalización. Esta función asigna un estado fijo (de A_c) para cada $c \epsilon C - C_p$. Sea Q una subfamilia de caracteres y α una función de asignación de estado de penalización. Una subfilogenia para (Q, α) es una filogenia C_p-perfecta T para Q en la que la raíz x cumple:

1. $\forall c \epsilon C_p$, $c(x) = c(Sv(Q))$ si $c(Sv(Q)) \neq *$; sino $c(x) = c(u)$ para algún $u \epsilon Q$.

2. $\forall c \epsilon C - C_p$, $c(x) = \alpha(c)$.

Dada una subfilogenia T con raíz x para (Q, α). Tomando (u, v) como una arista de T, donde u es el nodo padre de v, (u, v) es una arista buena si $S \cap V(T_v)$ es una subfamilia de caracteres; de lo contrario, (u, v) es una arista mala. El subárbol más grande de T que contiene a x y solo aristas malas es el árbol malo de T y se representa como $B(T)$. T está en forma normal si, para toda arista buena (u, v) de T tal que $u \epsilon V(B(T))$, T_v es una subfilogenia para algún par (Q_v, α_v), con $Q_v \subseteq Q$.

5.2.2. Algoritmos PP y NPP

En primer lugar se va a mostrar el algoritmo de construcción de filogenias perfectas, que se encuentra en la página 1120 en [17].

FILOGENIA_PERFECTA(S, C):

1. Generar una tabla N con una entrada por cada subfamilia Q, asignando $N(Q) = \emptyset$.

2. Enumerar, por cardinalidad creciente, cada subfamilia de caracteres Q, y ejecutar SUBFILOGENIA(Q).

SUBFILOGENIA(Q):

Para cada subfamilia $Q_1 \subset Q$ compatible con Q hacer:

a) $T_{Q_1} = N(Q_1)$.

b) Si $T_{Q_1} \neq \emptyset$, entonces:

I) $Q_2 = Q - Q_1$ y $T_{Q_2} = N(Q_2)$.

II) Si $T_{Q_2} \neq \emptyset$ entonces T_{Q_2} es una subfilogenia de Q cuya raíz es un nodo x_Q que satisface $c(x_Q) = c(Sv(Q, Q_1))$, $\forall c \epsilon C$ tal que $c(Sv(Q, Q_1)) \neq *$, y $c(x_Q) = c(x_{Q_1})$ para cualquier otro c, donde $c(x_{Q_1})$ es la raíz de T_{Q_1}. Asingar $N(Q) = T_Q$ y volver.

III) Por el contrario, sea $\{P_i\}_{i=1}^{k}$ el conjunto de clases de equivalencia $Q_2/Sv(Q, Q_1)$. Si $T_{P_i} = N(P_i) \neq \emptyset$, $\forall i \epsilon \{1, \ldots, k\}$, entonces se crea la subfilogenia T_Q de Q cuya raíz x_Q satisface $c(x_Q) = c(Sv(Q, Q_1))$, $\forall c \epsilon C$ y cuyos subárboles son T_{Q_1} y $T_{P_1}, \ldots, T_{P_k}$. Asingar $N(Q) = T_Q$ y volver.

---

3. Si existe un par de subfamilias Q_1, Q_2 tal que $Q_2 = S - Q_1$ y $N(Q_1)$, $N(Q_2) \neq \emptyset$, entonces devolver el árbol T obtenido como resultado de enlazar las raíces de $T_{Q_1} = N(Q_1)$ y $T_{Q_2} = N(Q_2)$ mediante una arista. En caso contrario, devolver $\emptyset$.

Una vez visto el funcionamiento del algoritmo para construcción de filogenias perfectas, a continuación se puede ver el algoritmo en el que se centra el trabajo de este capítulo: el algoritmo para construcción de filogenias casi perfectas, que se encuentra en la página 1123 en [17].

FILOGENIAS_CASI_PERFECTAS(S, C, q):

1. Se ejecuta FILOGENIA_PERFECTA(S, C). Si el algoritmo encuentra un resultado, devolvemos ese árbol como solución; sino, se pasa al siguiente punto del algoritmo.

2. Si $|S| \leq qr + 1$, entonces aplicar enumeración exhaustiva para buscar la filogenia casi perfecta de menor longitud con penalización q para (S, C). Devolver $\emptyset$ en caso de no existir tal filogenia. En caso contrario, se devuelve al filogenia.

3. Para cada $C_p \subseteq C$ tal que $|C_p| \geq m - q$, encontrar una filogenia C_p-perfecta de longitud mínima T_{C_p} con penalización, como mucho, q para S, si existe, de la siguiente forma:

   a) Generar una tabla N con una entrada por cada posible par (Q, α), con Q una subfamilia y α una función de asignación de estado de penalización. Establecer $N(Q, \alpha) = \emptyset$ para cada par (Q, α).

   b) Enumerar, por cardinalidad creciente, todos los pares (Q, α). Para cada (Q, α), aplicar SUBFILOGENIA(Q, α, q).

   SUBFILOGENIA(Q, α, q):

   Para encontrar la subfilogenia de longitud mínima de (Q, α), se generarán todos los candidatos posibles que permitan generar una subfilogenia. Si existen varias, se almacenará en la tabla de de menor longitud.

   Un candidato está compuesto por una posible topología de $\tilde{B}$, una función de asignación de estado de penalización para cada nodo v de $\tilde{B}$ (si v es la raíz entonces esta función ha de ser α), y subconjuntos de Q escogidos de una forma específica para asignarlos a cada nodo v de $\tilde{B}$. Para más detalles sobre este proceso ver el apartado 5.1 de [17].

   ---

   c) El árbol de mínima longitud T_{C_p} se obtiene al poner una arista entre las raíces de las subfilogenias para (Q_1, α_1) y (Q_2, α_2), tal que $Q_2 = Q - Q_1$ y $N(Q_1)$, $N(Q_2) \neq \emptyset$.

4. Devolver el árbol T_{C_p} que minimice $length\left(T_{C_p}\right)$ de entre todos los conjuntos C_p enumerados. Si no existe ningún árbol, devolver $\emptyset$.

5.3 Análisis de complejidad

Como el objetivo va a ser aplicar modificaciones sobre el algoritmo de construcción de filogenias casi perfectas, a continuación se va a mostrar su complejidad asociada de forma detallada. Esto permitirá no sólo identificar aquellas partes del algoritmo de mayor coste computacional, sino que también posteriormente facilitará evaluar las mejoras que se realicen.

En primer lugar, el coste temporal de encontrar una filogenia perfecta es $O\left(2^{2r}nm^2\right)$. Un análisis más minucioso de este coste se encuentra en [17]. Como es obvio, una filogenia perfecta es un caso particular de filogenia casi perfecta, por lo que la solución, de existir, es válida como solución al problema planteado.

El número total de conjuntos C_p posibles es $\sum_{i=m-q}^{m} \binom{m}{i} = O\left(qm^q\right)$ y se enumeran $O\left(m2^r r^q\right)$ pares (Q, α). Por otro lado, la generación de candidatos tiene varios costes asociados:

Topologías para el árbol $\tilde{B}$: existen $qr^{O(qr)}$ topologías distintas y el coste temporal de generarlos es el mismo.

Funciónα: existen $r^{O\left(q^2r\right)}$, que se pueden generar en un tiempo $2^{O\left(q^2r^2\right)}$.

Subfamilias de secuencias: existen $m^{O(q)}2^{O(qr)}$ posibilidades y $(qr)^{O(q)}$ formas de distribuirlas por los vértices de $\tilde{B}$.

En resumen, el número total de candidatos es $m^{O(q)}2^{O\left(q^2r^2\right)}$, que pueden ser generados en ese mismo coste temporal. Por otro lado, procesar un candidato tarda $O\left(|Q|r^q\right)$. El coste temporal de encontrar una subfilogenia con penalización mínima para (Q, α) es, por tanto, $|Q|m^{O(q)}2^{O\left(q^2r^2\right)}$.

En consecuencia, el algoritmo para la construcción de filogenias casi perfectas tiene un coste $|S|m^{O(q)}2^{O\left(q^2r^2\right)}$. Este problema, fijando q y r, puede ser resuelto en tiempo polinómico.

5.4 Propuesta

A continuación se van a exponer los cambios que se han aplicado hasta la fecha a los distintos elementos que influyen en el algoritmo y su coste computacional.

5.4.1. Modelo evolutivo

En primer lugar, dado que el criterio de MV requiere de un modelo evolutivo para poder realizar la evaluación, se va a definir un modelo evolutivo propio, más sencillo que la mayoría de modelos evolutivos utilizados en el Capítulo 3.

El modelo evolutivo que se contempla en la propuesta tiene una probabilidad P de que cualquier valor de cualquier posición de una secuencia permanezca invariable en la secuencia hija. Esta probabilidad, por estar tratando secuencias biológicas, tendrá que ser un valor superior a 0.5. Como es obvio, la probabilidad de que ocurra una mutación será $1 - P$, que será inferior a 0.5.

Por otro lado, es necesario establecer la probabilidad π de cada valor en cada una de las posiciones de la secuencia raíz del árbol (o subárbol). Esta probabilidad es igual a $\frac{1}{r_c}$, con $c \epsilon C$.

5.4.2. Función de penalización

La ecuación 5.1 muestra la fórmula de penalización que utiliza el algoritmo bajo el criterio de máxima parsimonia.

$$penalty\,(T) = length\,(T) - \sum_{c \epsilon C} (r_c - 1) \tag{5.1}$$

Con el cambio de criterio, una de las características importantes de este método es que cada posición de las secuencias se considera que evoluciona independientemente del resto de posiciones, por lo que se puede hacer un tratamiento por "columnas" (viendo el alineamiento como una matriz de valores). T_c representa el árbol en el que en cada nodo solo se tiene el valor σ de la secuencia correspondiente en esa columna c. Por lo tanto, la nueva función de penalización para cada posición se muestra en la ecuación 5.2.

$$penalty\,(T_c) = \log\left(L\,(PP_c)\right) - \log\left(L\,(T_c)\right) \tag{5.2}$$

Donde PP_c representa la filogenia perfecta (si existe) para esa columna. En las ecuaciones 5.3 y 5.4 se muestran las equivalencias para los cálculos de la verosimilitud en cada caso. $E\,(T)$ representa el conjunto de ramas del árbol T; l_c es el número de ramas de T_c en las que existe una mutación.

$$L\,(PP_c) = \pi P^{(|E(T)|+1-r_c)}\,(1-P)^{(r_c-1)} \tag{5.3}$$

$$L\,(T_c) = \pi P^{(|E(T)|-l_c)}\,(1-P)^{l_c} \tag{5.4}$$

Estos son los cálculos para cada columna, como ya se ha comentado. La función final de penalización para el criterio de máxima verosimilud se encuentra en la ecuación 5.5. Para los cálculos del siguiente apartado se ha utilizado esta misma función con el contenido del sumatorio simplificado (ecuación 5.6).

$$penalty\,(T) = \sum_{c \epsilon C} \left(\log\left(L\,(PP_c)\right) - \log\left(L\,(T_c)\right)\right) \tag{5.5}$$

$$penalty\,(T) = \sum_{c \epsilon C} \log\left(P^{(l_c+1-r_c)}\,(1-P)^{(r_c-1-l_c)}\right) \tag{5.6}$$

5.4.3. Lemas

A continuación se van a mostrar los lemas que han tenido que ser adaptados al nuevo criterio y función de penalización.

Lema 4: Sea T una filogenia C_p-perfecta tal que $penalty\,(T) \leq q$. Entonces T tiene, como máximo, $\left\lceil \frac{q}{\log\left(\frac{P}{1-P}\right)} + mr \right\rceil$ aristas malas.

Demostración: Sea $C' = C - C_p$. Para cada $c \epsilon C'$, sea l_c el número de aristas(u, v) en T tal que $c\,(u) \neq c\,(v)$. Por el Lema 3 [17], para cada arista mala (u, v) tiene que existir un $c \epsilon C'$ tal que $c\,(u) \neq c\,(v)$. Además, $1 \leq C' < m$, dado que T es una filogenia casi perfecta. Por lo tanto, el número de aristas malas es, como máximo, $\sum_{c \epsilon C'} l_c$. Por otro lado,

$$\sum_{c \epsilon C'} \log\left(P^{(l_c+1-r_c)}\,(1-P)^{(r_c-1-l_c)}\right) \leq q,$$

$$\log\left(\frac{P}{1-P}\right) \sum_{c \epsilon C'} l_c + |C'|\,(1-r) \log\left(\frac{P}{1-P}\right) \leq q,$$

$$\sum_{c \epsilon C'} l_c \leq \frac{q}{\log\left(\frac{P}{1-P}\right)} + |C'|\,(r-1) \leq \left\lceil \frac{q}{\log\left(\frac{P}{1-P}\right)} + mr \right\rceil$$

Por lo tanto, el número de aristas malas está limitado por $\left\lceil \frac{q}{\log\left(\frac{P}{1-P}\right)} + mr \right\rceil$.

Lema 5: Supongamos que Q es una subfamilia de caracteres la cual tiene una filogenia C_p-perfecta T, siendo x la raíz de T y que cumple que $\forall c \epsilon C_p$, $c\,(x) = c\,(Sv\,(Q))$ si $c\,(Sv\,(Q)) \neq *$. Sea α la asingación de penalización de x. Entonces, (Q, α) tiene una subfilogenia con un valor de verosimilitud, como máximo, igual al de T.

La demostración sigue siendo válida para el enunciado presentado.

Lema 6: Supongamos que el par (Q, α) tiene una subfilogenia. Entonces, (Q, α) tiene una subfilogenia de máxima verosimilitud en forma normal.

La demostración sigue siendo válida para el enunciado presentado.

6 Conclusiones

Se ha desarrollado un sistema donde se ha aplicado una metodología divide y vencerás para el diseño de flujos de trabajo (empleando el principio de transparencia de caja negra) que integra selección de modelos y reconstrucción filogenética, y que puede lidiar con análisis de filogenias extensivas de forma eficiente. El sistema ha sido probado con un conjunto grande de datos de mtDNA, aunque acepta cualquier tipo de dato biológico como entrada. Además, el criterio para la división de los datos de entrada en subconjuntos puede ser personalizado para reflejar correctamente la naturaleza los mismos; por supuesto, el número de muestras estadísticas también puede ser modificado. El sistema obtiene *speedups* mayores a 50 comparándolo con su equivalente secuencial en estudios filogenéticos grandes; se han obtenido grandes mejoras en la fase de selección de modelos comparándola de forma independiente con herramientas de objetivo específico como *jModelTest*.

Se ha presentado un nuevo algoritmo para evaluar los errores cometidos por el proceso de secuenciación, proporcionando como salida el nivel de veracidad de la secuencia dada una filogenia. Actualmente, esta comprobación de las secuencias se realiza de forma manual, lo que implica una gran inversión de tiempo, sin tener en cuenta los posibles errores humanos. La solución propuesta ofrece un detector automático de posibles errores cometidos en el proceso de secuenciación, con un rendimiento muy bueno, que también muestra como resultado los nodos más cercanos de la filogenia a la nueva secuencia. Si ésta se considera lo suficientemente buena, el algoritmo automáticamente la agrega a la filogenia, por la que la información está siempre actualizada.

Finalmente, como trabajo futuro del sistema se buscará obtener mejoras en el *speedup* conseguido, que parece degradarse a medida que el tamaño del problema crece. También se contemplarán formas de integrar, como tareas previas al sistema actual, la obtención automática de datos a partir de la entrada y el proceso de alineamiento de las secuencias. En este contexto, se espera mejorar los costes computacionales investigando nuevos criterios que permitan explotar más aún la naturaleza inherentemente paralela de este tipo de procesos.

En lo concerniente al algoritmo de verificación de secuencias, como fu-

turas mejoras se plantea desarrollar un nuevo proceso de verificación de las mutaciones detectadas como posibles errores, añadiendo un nuevo nivel de viabilidad biológica. Esta verificación consistirá en tener en cuenta las reversiones, por lo que una mutación que ya haya aparecido anteriormente en el árbol implicará que no se trata de una mala mutación. Además, el ratio de conservación entre distintas especies proporcionará un criterio extra para evaluar la viabilidad biológica de la mutación. Por otra parte, el proceso de alineamiento siempre es una tarea que implica un gran consumo de tiempo, así que cualquier mejora en esta fase mejorará el rendimiento global del algoritmo.

Respecto al trabajo ya planteado sobre complejidad paramétrica, resta finalizar la demostración de que la generalización al caso de MV propuesta tiene unas garantías de prestaciones y eficiencias razonables. Al tratarse de un trabajo de índole teórico pueden surgir dificultades adicionales en estas demostraciones, pero a su vez se puede contar con una evaluación experimental del algoritmo que se plantea. Por otro lado, se propone también extender el algoritmo a casos generales de máxima verosimilitud y máxima verosimilitud ancestral, además de abordar pruebas de completitud para los problemas cercanos para los que no se encuentren soluciones paramétricas eficientes.

Bibliografía

[1] Achilli, A., Rengo, C., Magri, C., Battaglia, V., Olivieri, A., Scozzari, R., Cruciani, F., Zeviani, M., Briem, E., Carelli, V., Moral, P., Dugoujon, J., Roostalu, U., Loogvöli, E., Kivisild, T., Bandelt, H., Richards, M., Villems, R., Santachiara-Benerecetti, A., Semino, O., and Torroni, A. The molecular dissection of mtDNA haplogroup H confirms that the Franco-Cantabrian glacial refuge was a major source for the European gene pool. *American Journal of Human Genetics 75* (Nov 2004), 910–918.

[2] Agarwala, R., and Fernández-Baca, D. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM Journal on Computing 23*, 6 (1994), 1216–1224.

[3] Álvarez-Jarreta, J. Análisis teórico-práctico de métodos de inferencia filogenética basados en selección de modelos y métodos de superárboles. Master's thesis, Centro Politécnico Superior, Universidad de Zaragoza, 2010.

[4] Andrews, R., Kubacka, I., Chinnery, P., Lightowlers, R., Turnbull, D., and Howell, N. Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA. *Nature Genetics 23* (Oct. 1999), 147.

[5] Bandelt, H., Macaulay, V., and Richards, M., Eds. *Human mitochondrial DNA and the evolution of Homo sapiens*. Springer, Berlin, Germany, 2006.

[6] Benson, D., Karsch-Mizrachi, I., Lipman, D., Ostell, J., and Sayers, E. GenBank. *Nucleic Acids Research 38* (Jan. 2010), D46–D51.

[7] Bininda-Emonds, O., Gittleman, J., and Steel, M. The (super)tree of life: procedures, problems and prospects. *Annual Review of Ecology and Systematics 33* (Dec. 2002), 265–289.

[8] Blanco, R., and Mayordomo, E. ZARAMIT: a system for the evolutionary study of human mitochondrial DNA. In *IWANN 2009, Part II* (2009), vol. 5518 of *Lecture Notes in Computer Science*, pp. 1139–1142.

[9] Blanco, R., Mayordomo, E., Montes, E., Mayo, R., and Alberto, A. *Advances in Bioinformatics*. Springer, Heidelberg, 2010, ch. Scalable phylogenetics through input preprocessing, pp. 123–130.

[10] Blanco, R., Mayordomo, E., Montoya, J., and Ruiz-Pesini, E. Rebooting the human mitochondrial phylogeny: an automated and scalable methodology with expert knowledge. *BMC Bioinformatics 12* (May 2011), 174.

[11] Bowers, S., McPhillips, T., Riddle, S., Anand, M., and Ludäscher, B. *Provenance and Annotation of Data and Processes*. Springer, Heidelberg, 2008, ch. Kepler/pPOD: scientific workflow and provenance support for assembling the tree of life, pp. 70–77.

[12] Couvares, P., Kosar, T., Roy, A., Weber, J., and Wenger, K. *Workflows for e-Science*. Springer, 2006, ch. Workflow management in Condor, pp. 357–375.

[13] Degnan, J., and Rosenberg, N. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in Ecology & Evolution 24* (Jun. 2009), 332–340.

[14] Edgar, R. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research 32* (Mar. 2004), 1792–1797.

[15] Felsenstein, J. *Inferring Phylogenies*. Sinauer Associates, 2004.

[16] Fernández-Baca, D. The perfect phylogeny problem. In *Steiner Trees in Industries* (2000), D. Du and X. Cheng, Eds., Kluwer Academic Publishers.

[17] Fernández-Baca, D., and Lagergren, J. A polynomial-time algorithm for near-perfect phylogeny. *SIAM Journal on Computing 32* (2003), 1115–1127.

[18] Georgakopoulos, D., Hornick, M., and Sheth, A. An overview of workflow management: from process modeling to workflow automation infrastructure. *Distributed and Parallel Databases 3*, 2 (1995), 119–153.

[19] Gramm, J., Nickelsen, A., and Tantau, T. Fixed-parameter algorithms in phylogenetics. *The Computer Journal* (2007).

[20] Herrnstadt, C., Elson, J., Fahy, E., Preston, G., Turnbull, D., Anderson, C., Ghosh, S., Olefsky, J., Beal, M., Davis, R., and Howell, N. Reduced-median-network analysis of complete mitochondrial DNA coding-region sequences for the major African, Asian, and European haplogroups. *American Journal of Human Genetics 70*, 5 (2002), 1152–1171.

[21] Holder, M., and Lewis, P. Phylogeny estimation: traditional and bayesian approaches. *Nature Reviews Genetics 4* (2003), 275–284.

[22] Hüffner, F., Niedermeier, R., and Wernicke, S. Techniques for practical fixed-parameter algorithms. *The Computer Journal 51*, 1 (2008).

[23] Kim, S., Tang, H., and Mardis, E., Eds. *Genome sequencing technology and algorithms*. Artech House, Norwood, MA, 2007.

[24] Luscombe, N., Greenbaum, D., and Gerstein, M. What is bioinformatics? an introduction and overview. In *Yearbook of Medical Informatics 2001* (2001).

[25] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M., Wipat, A., and Li, P. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics 20*, 17 (2004), 3045–3054.

[26] Olsen, G., Matsuda, H., Hagstrom, R., and Overbeek, R. fastdnaml: a tool for construction of phylogenetic trees of dna sequences using maximum likelihood. *Computer Applications in the Biosciences 10* (1994), 41–48.

[27] Piontkivska, H. Efficiencies of maximum likelihood methods of phylogenetic inferences when different substitution models are used. *Molecular Phylogenetics and Evolution 31* (2004), 865–873.

[28] Polanski, A., and Kimmel, M. *Bioinformatics*. Springer, 2007.

[29] Porter, M., Pérez-Losada, M., and Crandall, K. Model-based multi-locus estimation of decapod phylogeny and divergence times. *Molecular Phylogenetics and Evolution 37* (2005), 355–369.

[30] Porter, M., Pérez-Losada, M., and Crandall, K. Advantages of multilocus sequence analysis for taxonimic studies: a case study using 10 housekeeping genes in the genus *ensifer* (including former *sinorhizobium*). *International Journal of Systematic and Evolutionary Microbiology 58* (2008), 200–214.

[31] Posada, D. jModelTest: phylogenetic model averaging. *Molecular Biology and Evolution 25*, 7 (2008), 1253–1256.

[32] Posada, D., and Buckley, T. Model selection and model averaging in phylogenetics: Advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Systematic Biology 53* (2004), 793–808.

[33] Rajkumar, R., Banerjee, J., Gunturi, H., Trivedi, R., and Kashyap, V. K. Phylogeny and antiquity of M macrohaplogroup inferred from complete mt DNA sequence of Indian specific lineages. *BMC Evolutionary Biology 5* (Apr. 2005), 26.

[34] Richards, M., Macaulay, V., Bandelt, H., and Sykes, B. Phylogeography of mitochondrial DNA in western Europe. *Annals of Human Genetics 62*, 3 (1998), 241–260.

[35] Ruiz-Pesini, E., Lott, M., Procaccio, V., Poole, J., Brandon, M., Mishmar, D., Yi, C., Kreuziger, J., Baldi, P., and Wallace, D. An enhanced mitomap with a global mtdna mutational phylogeny. *Nucleic Acids Research 35* (2007), D823–D828.

[36] Ruiz-Pesini, E., Mishmar, D., Brandon, M., Procaccio, V., and Wallace, D. Effects of purifying and adaptive selection on regional variation in human mtdna. *Science 303* (2004), 223–226.

[37] Salas, A., Lareu, V., Calafell, F., Bertranpetit, J., and Carracedo, A. mtdna hypervariable region ii (hvii) sequences in human evolution studies. *European Journal of Human Genetics 8* (2000), 964–974.

[38] Sanderson, M., Purvis, A., and Henze, C. Phylogenetic supertrees: assembling the trees of life. *Trends in Ecology and Evolution 13*, 3 (1998), 105–109.

[39] Stamatakis, A., Ludwig, T., and Meier, H. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics 21* (2005), 456–463.

[40] Steel, M. The maximum likelihood point for a phylogenetic tree is not unique. *Systematic Biology 43* (1994), 560–564.

[41] Strimmer, K. *Maximum Likelihood Methods in Molecular Phylogenetics.* PhD thesis, Ludwig-Maximilians-Universität München, 1997.

[42] Sullivan, J., and Joyce, P. Model selection in phylogenetics. *Annual Review of Ecology, Evolution, and Systematics 36* (2005), 445–466.

[43] Torroni, A., Achilli, A., Macaulay, V., Richards, M., and Bandelt, H. Harvesting the fruit of the human mtDNA tree. *Trends in Genetics 22*, 6 (2006), 339–345.

[44] van Oven, M., and Kayser, M. Updated comprehensive phylogenetic tree of global human mitochondrial DNA variation. *Human Mutation 29* (Feb. 2008), E386–E394.