Máster en Técnicas Estadísticas - Proyecto Fin de Máster

Cadenas de Markov en la Investigación del Genoma

Autora: *Sara Prada Alonso*

Directora de proyecto: *María de los Ángeles Casares de Cal*

8 de Julio de 2013

Agradecimientos

En primer lugar, me gustaría expresar mi más sincero agradecimiento a mi directora de proyecto, María de los Ángeles Casares de Cal, por haber confiado en mí como alumna para realizar el presente trabajo, y haberme dado todas las facilidades para poder desarrollarlo a lo largo de estos meses. Agradezco encarecidamente su ayuda, interés, dedicación y labores de supervisión.

Del mismo modo, quiero darle las gracias a Antonio Gómez Tato por ayudarme a comprender mejor la materia de estudio y sus aplicaciones, especialmente en el campo computacional. Le agradezco igualmente y de forma encarecida su labor de supervisión y soporte en la búsqueda de cierta información.

Todos las materias cursadas durante el Máster me han aportado los conocimientos necesarios para comprender mejor y más rápidamente nuevos temas a abordar, como el que presento en esta memoria. Gracias a mis profesores y a la formación personal y profesional que me han aportado.

Por último, agradecer a mi familia y amigos su apoyo incondicional en este tiempo de trabajo.

Resumen

Los modelos de Markov ocultos están siendo utilizados actualmente para modelizar familias de proteínas en la búsqueda automática de genes, y en el alineamiento múltiple de secuencias, entre otras aplicaciones.

Existen varios algoritmos para estimar los parámetros de estos modelos, a partir de los datos. Se trata de hacer una revisión de los mismos y aplicarlo a datos reales.

Índice general

Capítulo 1

Introducción general

La aparición de los secuenciadores automáticos y sus desarrollos posteriores ha puesto a disposición de los investigadores ingentes cantidades de secuencias de ADN que necesitan ser analizadas. Una de las primeras tareas consiste en identificar las diferentes "regiones" de la secuencia y su anotación posterior (estimar su función en comparación con lo encontrado en otras especies).

Es conocido que la mayor parte del genoma humano es "no codificante" (no da lugar a un gen) aunque está cada vez más claro que esas regiones, antaño denominadas "ADN basura", contienen información relevante y son determinantes a la hora de comprender el comportamiento de la célula. La Bioinformática se encarga, entre otras cosas, del diseño de herramientas computacionales (utilizando técnicas matemáticas) para el discernimiento de la estructura del ADN y la identificación de las diferentes partes del mismo. Los métodos empleados son muy diversos, pero entre ellos destaca el uso de modelos probabilísticos más o menos complejos para intentar encontrar y clasificar las diferentes regiones, como pueden ser genes, pseudogenes, micro ARNs, regiones promotoras, intrones, exones, regiones CpG, etc. Entre los modelos probabilísticos más usados están los **modelos de Markov ocultos**, originalmente introducidos en el análisis de textos, pero que están siendo utilizados con éxito en Bioinformática y Filogenia Molecular, por citar dos disciplinas relacionadas o que utilizan información de secuencias de ADN.

La molécula de ADN se codifica por una secuencia de letras (A, C, G, T) que resume su estructura química. Un primer modelo probabilístico consistiría en utilizar la distribución multinomial, es decir, que la probabilidad de aparición de una secuencia de longitud n seguiría una distribución multinomial con parámetros n, p_A, p_C, p_G y p_T ($p_A + p_C + p_G + p_T = 1$), lo cual significaría que la aparición de un nucleótido en una posición determinada sería independiente de las demás. Pero éste no será un buen modelo, ya que no tiene en cuenta la información que aportan los nucleótidos adyacentes. Para incorporar esa información, puede utilizarse un modelo de Markov que determine la probabilidad de aparición de un nucleótido en una posición $i+1$ en función de lo que ocurre en la posición i de la cadena (o incluso en

las posiciones $i-k, i-(k+1), \ldots, i$. Este modelo funcionaría si toda la secuencia se comportase de manera semejante, pero no es así; existen "regiones" diferentes a lo largo del genoma, por ejemplo, regiones ricas en dinucleótidos CG, llamadas *islas CpG*. En esos casos, la probabilidad de aparición de un nucleótido en una posición (sitio) i de la cadena ya no sólo depende del nucleótido o nucleótidos anteriores sino que también influye el "estado" o la región que lo precede. Esos estados están ocultos y de ahí la necesidad de modelizar o utilizar modelos de probabilidad como los "modelos de Markov ocultos".

Desde sus primeras aplicaciones hasta la actualidad, los modelos de Markov ocultos (HMM: Hidden Markov Models) han ido aumentando su complejidad. Veremos someramente uno de los que se utilizan para la determinación o búsqueda automática de genes mediante uno de los programas más utilizados, *GENSCAN*, cuyo estudio queda fuera del objetivo de esta memoria.

Como ejemplo de aplicación de los modelos de Markov ocultos estudiaremos el problema de la determinación de las regiones ricas en CG (islas CpG). Debido al proceso biológico de la *metilación* (adición de un grupo metilo (-CH3) a una molécula), existe una probabilidad bastante alta de que esta metilación de C mute en una T, con la consecuencia de que, en general, los dinucleótidos CpG son más raros en el genoma de lo que podría esperarse de las probabilidades independientes de C y G. Debido a importantes razones biológicas, el proceso de metilación se suprime en determinadas extensiones cortas del genoma, como alrededor de los "promotores" o regiones de inicio de muchos genes. En estas regiones, observamos más dinucleótidos CpG que en ningún otro lugar, y de hecho más nucleótidos C y G en general. Tales regiones se denominan *islas CpG* y, en contraste, el resto del genoma es el "océano". Normalmente están compuestas por cientos de bases de longitud.

Una primera aproximación al problema anterior consiste en ver si una región determinada es o no una isla CpG. Para ello, basta utilizar un test de razón de verosimilitudes. Una etapa posterior consistirá en utilizar modelos de Markov ocultos muy sencillos, como puede ser el del "casino deshonesto", que no tiene en cuenta la información de los nucleótidos precedentes. Existen modelos de Markov ocultos sofisticados también para atacar el problema, uno de ellos ha aparecido recientemente en el artículo de Irizarry et al. ([24]). Nosotros nos detendremos en la aplicación del modelo de Markov oculto más utilizado actualmente para estudiar el tema.

Una vez escogido el modelo y estimados sus parámetros mediante una secuencia o secuencias "de entrenamiento", se necesita un algoritmo para aplicarlo y determinar las diferentes regiones de una nueva secuencia. Uno de los algoritmos más utilizados es el algoritmo de Viterbi, que encuentra el "camino óptimo" mediante técnicas de programación dinámica. Como ejemplo, hemos estudiado el uso de los modelos de Markov ocultos en el cromosoma 10 del genoma del "Danio rerio" o "pez cebra" (uno de los "genomas-modelo" estudiados en la actualidad). Hemos descargado la secuen-

cia del cromosoma 10 así como las regiones ya reconocidas como islas, tal como están en las bases de datos dedicadas a estas áreas; y utilizando el algoritmo de Viterbi, programado en "R", hemos estimado la aparición de regiones "CpG".

Terminamos esta memoria con un par de notas sobre algunos de los casos más recientes del uso en Bioinformática de los modelos de Markov ocultos.

Capítulo 2

Nociones básicas de biología

La molécula de ADN de cada organismo está formada por dos hebras de nucleótidos. Existen cuatro tipos distintos de nucleótidos, cuyas bases son: Adenina, Timina, Guanina y Citosina. La Adenina sólo puede unirse a la Timina y la Guanina a la Citosina, por eso decimos que la Adenina es *complementaria* de la Timina; y la Citosina es *complementaria* de la Guanina. Cada nucleótido de una cadena está unido al nucleótido que se encuentra enfrente en la otra cadena, y las dos hebras están plegadas formando una "doble hélice". Los cuatro nucleótidos descritos forman largas secuencias de un "alfabeto" de cuatro letras A, T, G y C (para Adenina, Timina, Guanina y Citosina; respectivamente), llamado *código genético*. Estas secuencias experimentan cambios dentro de una población a lo largo de varias generaciones, como las mutaciones aleatorias que surgen y se mantienen en la población. Por lo tanto, dos secuencias bastante diferentes pueden derivar de un antepasado común.

Supongamos que tenemos dos pequeñas secuencias de ADN como las mostradas a continuación, quizás de dos especies distintas. Las letras en negrita indican parejas de nucleótidos que son el mismo en ambas secuencias.

$$\mathbf{G}\,G\,\mathbf{A}\,G\,A\,\mathbf{C}\,T\,G\,\mathbf{T}\;\mathbf{A}\;\mathbf{G}\,A\,\mathbf{C}\;\mathbf{A}\,G\,C\,T\,A\,A\,T\,G\,C\,\mathbf{T}\;\mathbf{A}\;\mathbf{T}\,A$$
$$\mathbf{G}\,A\,\mathbf{A}\,C\,G\,\mathbf{C}\,C\,C\,\mathbf{T}\;\mathbf{A}\;\mathbf{G}\,C\,\mathbf{C}\;\mathbf{A}\,C\,G\,A\,G\,C\,C\,C\,T\,\mathbf{T}\;\mathbf{A}\;\mathbf{T}\,C$$

Deseamos medir si las dos secuencias muestran una similitud significativa, para evaluar, por ejemplo, si provienen de un ascendiente común.

Supongamos una variable aleatoria X que mida el número de coincidencias en un conjunto de $n = 26$ observaciones, o letras de nuestro alfabeto en este caso. Si las secuencias son generadas aleatoriamente, teniendo las cuatro letras A, T, G y C igual probabilidad $p = \frac{1}{4}$ de ocurrir en cualquier posición, y consideramos que una coincidencia entre observaciones de ambas secuencias es un "éxito", tendríamos que X es una variable aleatoria que sigue una distribución Binomial de parámetros $n = 26$ y $p = \frac{1}{4}$.

$$X \backsim B(n,p)$$

Así, las dos secuencias deberían coincidir en un cuarto de las posiciones. Las dos secuencias anteriores coinciden en 11 de 26 posiciones, ¿cómo de improbable es este resultado si las secuencias son generadas aleatoriamente? La teoría de la probabilidad junto con lo expuesto en el párrafo anterior muestran que, asumiendo probabilidades iguales para A, T, G y C en cualquier posición, e independencia de todos los nucleótidos involucrados; la probabilidad de que haya 11 o más coincidencias en una comparación de secuencias de longitud 26 es aproximadamente de 0.04, tal y como se indica a continuación

$$\begin{aligned} P(X \geq 11) &= 1 - P(X < 11) = \\ &= 1 - (P(X=0) + P(X=1) + \ldots + P(X=10)) = 0.0400845, \end{aligned}$$

siendo

$$P(X=i) = \binom{26}{i} \left(\frac{1}{4}\right)^i \left(1 - \frac{1}{4}\right)^{26-i}, \qquad i = 0, 1, 2, \ldots, 26.$$

Por lo tanto, nuestra observación de 11 coincidencias muestra evidencias significativas de que esto no sólo es cosa del azar. Estudiaremos el modelo que se encuentra bajo las secuencias de nucleótidos, y que da lugar con una cierta probabilidad a unas u otras cadenas. A éste lo llamaremos *Modelo de Markov Oculto*.

Mostramos a continuación con más detalles algunas nociones básicas de biología que se necesitarán para comprender mejor las explicaciones posteriores ([40]).

El *ácido desoxirribonucleico* (ADN) es la información macromolecular básica de la vida. Consiste en un polímero de nucleótidos, en el cual cada nucleótido está compuesto por un azúcar (la desoxirribosa) y un grupo fosfato, conectado a una base nitrogenada de uno de los cuatro posibles tipos: adenina, guanina, citosina o timina (abreviadas como A, G, C y T, respectivamente). Los nucleótidos adyacentes en una sola hebra de ADN están conectados por un enlace químico entre el azúcar de uno de ellos y el grupo fosfato del siguiente. La estructura clásica de "doble hélice" del ADN se forma cuando dos hebras de ADN forman enlaces hidrogenados entre sus bases nitrogenadas, resultando en la ya familiar estructura de escalera. Bajo condiciones normales, estos enlaces hidrogenados se forman solo entre particulares pares de nucleótidos (en referencia a una de las parejas de bases): adenina se empareja solamente con timina, y guanina con citosina. Dos hilos de ADN son *complementarios* si la secuencia de bases en cada uno es tal que se emparejan correctamente a lo largo de toda la longitud de ambas cadenas. La secuencia en la cual las diferentes bases ocurran en una particular hebra de ADN representa la información genética cifrada en esa hebra. En virtud de la especificidad del emparejamiento de nucleótidos, cada una de las dos hebras de cualquier molécula de ADN contiene toda la información presente en la otra. Hay también una polaridad química en las cadenas

de polinucleótidos de modo que la información contenida en las bases A, G, C y T es sintetizada y decodificada en cualquier dirección. La molécula de ADN puede ser circular o lineal, y puede estar compuesta de 10,000 mil millones de nucleótidos en una cadena larga.

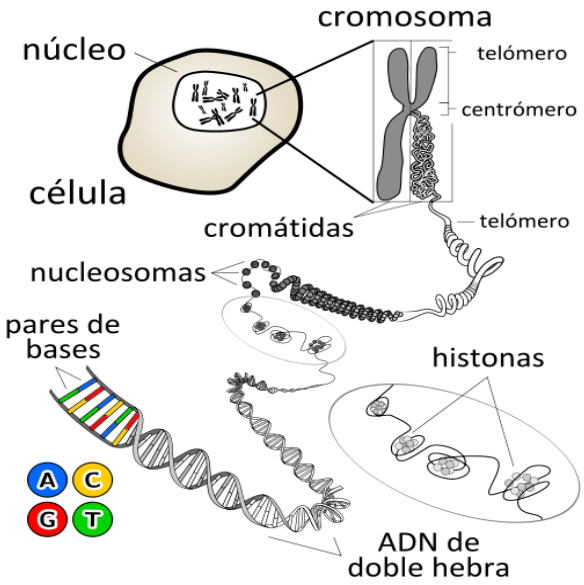

Figura 2.1: Situación del ADN dentro de la célula.

En la célula, el ADN se organiza en cromosomas, que se encuentran en el núcleo de ésta. Se trata de una sola pieza de espiral de ADN que contiene muchos genes, elementos reguladores y otras secuencias de nucleótidos. Los cromosomas varían ampliamente entre los diferentes organismos. Las células humanas contienen 23 pares de cromosomas, un miembro de cada par heredado de la madre y el otro del padre. Los dos cromosomas en un par son prácticamente idénticos, con la excepción del cromosoma sexual, para el cual hay dos tipos, X e Y (XX para la mujer y XY para el hombre, siendo la pareja la que determina el sexo). Cada célula del cuerpo contiene copias idénticas del conjunto entero de 23 pares de cromosomas. Como curiosidad, señalar que si se desenrollaran y pusieran en fila los cromosomas en cada una de las células (un humano adulto tiene entre 10 y 50 billones de células) la longitud total de ADN sería de unos 2 metros. Si se sumara la longitud del ADN de todas las células de una sola persona, se podría rodear la circunferencia terrestre 500,000 veces.

El conjunto total del ADN de un organismo es el *genoma*; que contiene más de tres billones de pares de bases. El genoma humano o secuencia completa de ADN de un organismo constituye la información genética heredable del núcleo celular. Un gen es un trozo de ADN que lleva la información para que se fabrique una proteína, que puede hacerse de múltiples formas. Los genes están en el núcleo de las células y las proteínas que codifican, que son las que controlan los caracteres, se fabrican

en el citoplasma de la célula (parte entre en núcleo y la membrana que delimita la célula). Existen, por tanto, moléculas intermediarias que pueden "copiar" un trozo de la cadena de ADN, atravesar la membrana del núcleo y ya en el citoplasma traducir la información almacenada en el ADN. Son las moléculas de *ARN mensajero*.

Un cromosoma humano está compuesto principalmente del llamado *ADN basura* o *ADN intergénico*, cuya función no está del todo comprendida. El ADN intergénico es ADN que no codifica proteínas. Ejemplos de éste son los *intrones*, que son segmentos internos dentro de los genes y que se eliminan a nivel de ARN; o los *pseudogenes*, inactivados por una inserción o supresión. Intercalados con estas áreas de ADN están los genes. Éstos se organizan en *exones*, que son las secuencias que serán eventualmente utilizadas por la célula, alternados con los intrones, que serán desechados. Actualmente se piensa que el genoma humano contiene aproximadamente entre 30,000 y 40,000 genes. La información en estos genes se transcribe en ARN (*ácido ribonucleico*), y en muchos casos finalmente en proteínas.

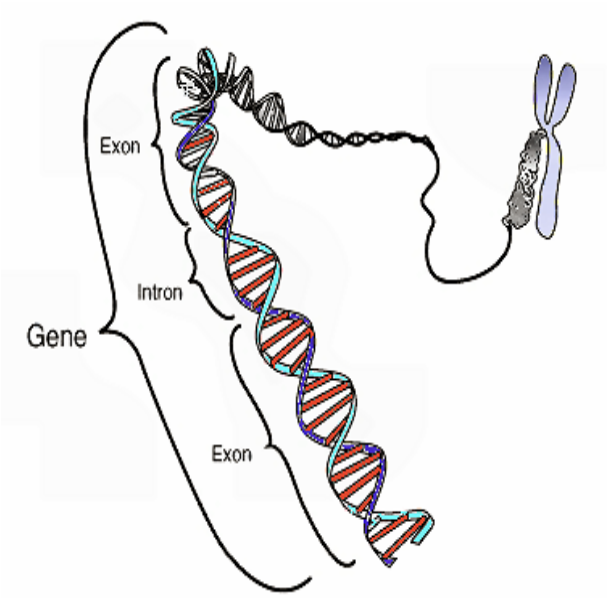

Figura 2.2: Situación del gen dentro del cromosoma.

La estructura de los genes y el mecanismo de expresión de éstos en organismos eucariotas es mucho más complicado que en los procariotas. En organismos eucariotas, la región de ADN codificante en proteína no es continua normalmente. Esta región está compuesta por extensiones de exones e intrones. Durante la transcripción, ambos son transcritos en ARN, en su orden lineal. Después de eso, tiene lugar un proceso llamado "de empalme" o "de ayuste" (*splicing*), en el cual las secuencias de intrones se extraen y se descartan de la secuencia de ARN. Los segmentos restantes de ARN, los que corresponden a los exones, se ligan para formar la hebra madura de ARN. Un gen multi-exón típico tiene la siguiente estructura: empieza con la *región promotora*, a la que le sigue una región transcrita pero no codificante llamada

región no traducida 5' (5'UTR: 5' Untranslated Region). Le sigue el exón inicial que contiene el codón de inicio (secuencia de grupos de tres nucleótidos que marcan el inicio del exón). Siguiendo al exón inicial, existen series alternadas de intrones y exones internos, seguidos por el exón final, que contiene el codón de terminación o final (secuencia de grupos tres nucleótidos que "señalan" el final del exón). Después, nos encontramos otra región no codificante llamada *región no traducida 3'* (3'UTR). Terminando el gen eucariota, hay una señal de poliadenilación (Poli(A): el nucleótido Adenina se repite varias veces). Los límites exón-intrón (es decir, los lugares de empalme) están señalados por cortas secuencias específicas (2 pares de bases de longitud). Una 5'UTR (3'UTR) terminada en un intrón (exón) se llama *sitio donor* (*donor site*), y una 3'UTR (5'UTR) que termine con un intrón (exón) se llama *sitio aceptor* (*acceptor site*).

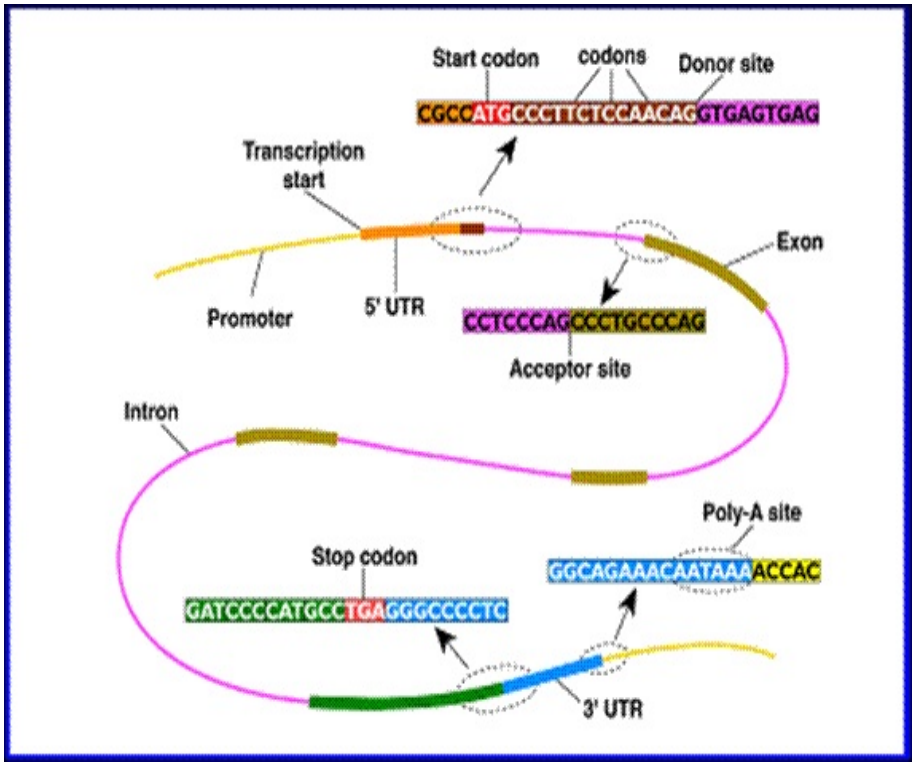

Figura 2.3: Estructura de un gen.

El primer paso en el proceso de conversión del gen a proteína es la *transcripción*, la creación de la molécula de ARN utilizando la secuencia de ADN del gen como plantilla. La transcripción se inicia en las secuencias no codificantes o *promotoras*, localizadas inmediatamente antes del gen. Como el ADN, el ARN se compone de series de nucleótidos, pero con varias diferencias importantes: el ARN está compuesto por una sola hebra, que contiene el "azúcar ribosa", y sustitutos de la base nitrogenada *Uracilo* por Timina. El Uracilo es una de las cuatro bases nitrogenadas que forman parte del ARN y en el código genético se representa con la letra U. Después de la transcripción, que incluye la eliminación de intrones, el ARN irá a diferentes

destinos dentro de la célula. De interés particular es el ARNm (*ARN mensajero*), que será convertido en proteína. Una proteína está compuesta de una secuencia de aminoácidos, que se unirán en el orden indicado en el "mensaje" que se encuentra en el ARN mensajero. Hay 20 aminoácidos que forman las proteínas. Cada uno de esos aminoácidos está representado por una o más secuencias de tres ARN nucleótidos conocidos como un *codón*; por ejemplo, la secuencia de ARN **AAG** codifica el aminoácido *lisina*. La lisina es uno de los 10 aminoácidos esenciales para los seres humanos. La combinación de cuatro posibles nucleótidos en grupos de tres da lugar a $VR_{4,3} = 4^3$ o 64 codones, es decir, que la mayoría de los aminoácidos están codificados por más de un codón. El *ribosoma* realiza el cambio del ARNm en proteína. Los ribosomas son complejos macromoleculares de proteínas y ácido ribonucleico (ARN) que se encuentran en el citoplasma y se encargan de sintetizar proteínas a partir de la información genética que les llega del ADN transcrita en forma de ARN mensajero (ARNm). El ribosoma empareja cada codón en la secuencia de ARN con el apropiado aminoácido, y después añade el aminoácido sobre la proteína en creación. El proceso de cambio está mediado por dos tipos especiales de codón: el *codón de inicio* señala la ubicación en el ARN molecular donde la traducción (o síntesis de proteínas) debe comenzar, mientras que los *codones finales* señalan el lugar donde la traducción debe terminar. Una vez que se compone la secuencia de aminoácidos, se "monta" una particular proteína, ésta se disocia del ribosoma y se pliega en una específica forma tridimensional. La función de la proteína depende fundamentalmente de su estructura tridimensional y su secuencia de aminoácidos. Las proteínas pasan a llevar a cabo una variedad de funciones en la célula, cubriendo todos los aspectos de las funciones celulares del metabolismo.

Actualmente, estas funciones celulares han sido asignadas a solamente una pequeña proporción de los genes, incluso en los modelos de organismos mejor conocidos. Para poder asignar funciones a los genes restantes, es útil examinar "expresiones patrones" (o repeticiones de secuencias de nucleótidos) de estos genes en varios tejidos. La tecnología "microarray" desarrollada en los años pasados, permite la medida de los niveles de ARNm para decenas de miles de genes simultáneamente. Esto proporciona una eficiente y buena manera de determinar la "expresión patrón" (o secuencia repetida) de los genes en muchos tipos diferentes de tejidos, pero al mismo tiempo proporciona nuevos retos en información catalogada y análisis estadístico.

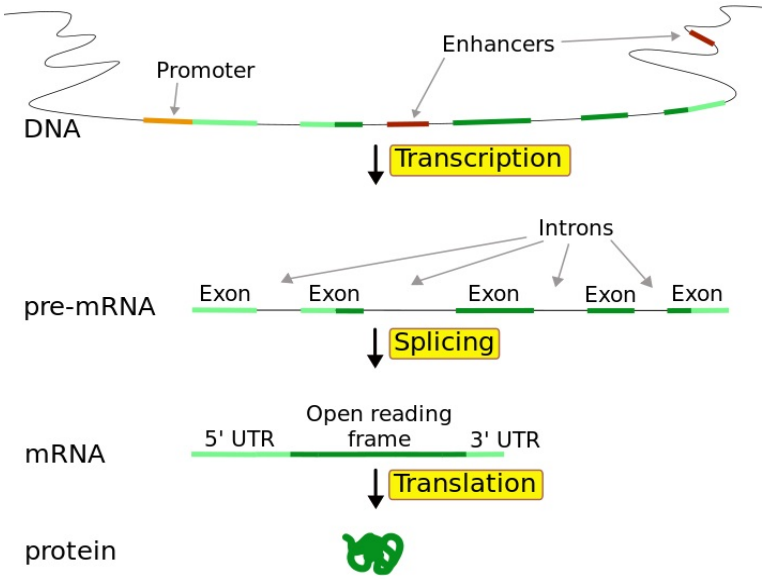

Figura 2.4: Proceso de creación de la proteína.

Capítulo 3

Introducción a las Cadenas de Markov

3.1. Introducción a las Cadenas de Markov Finitas

En este capítulo abordaremos una breve introducción sobre procesos estocásticos, en particular sobre las cadenas de Markov en tiempo discreto y finitas, repasando sus características y propiedades. El objetivo es introducir el material necesario para la comprensión de los modelos que desarrollaremos posteriormente.

Un *proceso estocástico* es un conjunto de variables aleatorias (estocásticas) $\{X_t, t \in T\}$. El parámetro o índice t se interpreta generalmente como "el tiempo", y los posibles valores de la v.a. X_t se interpretan como los posibles "estados" del proceso en el tiempo t. El conjunto de parámetros T puede ser continuo o discreto. Si T es discreto diremos que el proceso es *de parámetro o de tiempo discreto*. Si T es un intervalo de la recta real, entonces diremos que el proceso es *de parámetro o tiempo continuo*. Cada una de las variables aleatorias del proceso tiene su propia función de distribución de probabilidad y, entre ellas, pueden estar correlacionadas o no. Cada variable o conjunto de variables sometidas a influencias o impactos aleatorios constituye un proceso estocástico. Típicamente, describen algún fenómeno que evoluciona en el tiempo o en el espacio.

Dentro de los procesos estocásticos, encontramos los *procesos de Markov*, que son aquellos procesos discretos en los que la evolución en el tiempo (generaciones) o en el espacio (secuencias biológicas) sólo depende del estado actual y no de los anteriores. Un proceso de Markov, llamado así por el matemático ruso Andréi Markov, es un fenómeno aleatorio dependiente del tiempo para el cual se cumple una propiedad específica: la *propiedad de Markov*. En una descripción común, la condición de Markov, o "sin memoria", exige que la propiedad de que la cadena de Markov que se encuentre en un estado j en el instante $(n + 1)$ dependa únicamente del estado

en que se encontraba en el instante n, y que esto se cumpla para cualquier etapa en la que se encuentre la cadena. Frecuentemente el término *cadena de Markov* se utiliza para dar a entender que un proceso de Markov tiene un espacio de estados discreto (finito o numerable), es decir, es un proceso de Markov en tiempo discreto. Las Cadenas de Markov son muy útiles en Bioinformática, ya que se utilizan en la modelización de los procesos evolutivos de las cadenas de ADN.

De manera formal, diríamos que

Definición 3.1 *Una Cadena de Markov es una sucesión de variables aleatorias* $\{X_n, n \in \mathbb{N}\}$ *que cumple:*

1. *Cada variable* X_n *toma valores en un conjunto finito o numerable* E*, que se denomina espacio de estados.*

2. *La sucesión de variables verifica la condición o propiedad de Markov o markoviana:*

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, \ldots, X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n), \qquad (3.1)$$

donde $i_0, \ldots, i_n$ *denotan los estados en los que se encuentra la cadena en cada etapa.*

3.2. Probabilidades de transición y Matriz de transición

Las probabilidades condicionadas

$$P(X_{n+1} = j | X_n = i), \qquad (3.2)$$

son conocidas como las *probabilidades de transición*. La condición de *homogeneidad en el tiempo*, según la cual la probabilidad de pasar de i a j es independiente de la etapa en la que se encuentre la cadena, hace que la *probabilidad de transición de* i *a* j sea

$$p_{ij} = P(X_{n+1} = j | X_n = i) = P(X_{m+1} = j | X_m = i), \qquad \forall n, m \in \mathbb{N}\,; i, j \in E \qquad (3.3)$$

que indica la probabilidad de pasar al estado j desde el estado i, en cualquier etapa. Una Cadena de Markov en tiempo discreto (CMTD) que cumpla esta condición de homogeneidad en el tiempo (cadena *homogénea*) se dice que tiene probabilidades de transición *estacionarias*.

Supongamos que el espacio de estados E que manejamos, es decir el conjunto de posibles valores que puede tomar el proceso en cada una de sus etapas, tiene un número finito k de estados. En este caso, decimos que la cadena de Markov es *finita*.

Así, las probabilidades de transición entre estados, p_{ij}, se combinan formando una *matriz de transición* P de tamaño $k \times k$. Dada la homogeneidad de la cadena, la matriz es de la forma

$$P = (p_{ij}) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kk} \end{pmatrix}. \tag{3.4}$$

Los elementos de esta matriz P verifican que $p_{ij} \geqslant 0$, $\forall i, j \in E$. Además, dado que el estado es i en el instante n y que el proceso debe estar en algún estado en el instante $n+1$, se verifica que $\sum_{j \in E} p_{ij} = 1$, $\forall i \in E$; es decir, las sumas por filas son uno. Esta propiedad es característica de las *matrices estocásticas*, a las que pertenecen las matrices de transición.

Definición 3.2 *Sea P una matriz cuadrada $k \times k$. Llamaremos a P matriz estocástica si*

1. *Todas sus entradas son positivas, $pij \geq 0$, para todo i, j.*
2. *La suma de sus coeficientes en cada fila vale 1, es decir*

$$\sum_{j=1}^{k} p_{ij} = 1, \qquad \forall i = 1, \ldots, k \tag{3.5}$$

y doblemente estocástica si además la suma de sus coeficientes en cada columna también vale 1, o sea

$$\sum_{i=1}^{k} p_{ij} = 1, \qquad \forall j = 1, \ldots, k \tag{3.6}$$

Se tiene que las potencias de una matriz estocástica también lo son ya que, si U es el vector columna $k \times 1$ cuyos elementos son todos unos, y P' es otra matriz estocástica, entonces

1. PP' tiene todos los elementos ≥ 0, pues ambas los tienen por definición de matriz estocástica.
2. $(PP')U = P(P'U) = PU = U$, la suma de los elementos de cada fila es igual a la unidad ya que $PU = U$. La primera igualdad se obtiene por la propiedad asociativa de la multiplicación para las matrices, y la segunda y tercera por definición de matriz estocástica para P' y P, respectivamente.

Por lo tanto, podemos decir que cada fila de la matriz de transición es una distribución de probabilidad.

3.3. Distribución inicial

Una cadena de Markov está completamente determinada por la distribución de probabilidad del estado inicial X_0 y por las probabilidades de transición p_{ij}. La distribución inicial de la cadena se expresa en forma de vector, en el que cada componente indica la probabilidad de que la cadena se encuentre en el estado i en el instante inicial. De esta forma, se conoce el punto de partida del proceso. Se expresa como

$$P^{(0)} = (p_1^{(0)}, \ldots, p_i^{(0)}, \ldots, p_k^{(0)}), \tag{3.7}$$

donde $p_i^{(0)} = P(X_0 = i)$, $i \in E$.

Además, la distribución inicial cumple: $p_i^{(0)} \geq 0$ y $\sum_{i \in E} p_i^{(0)} = 1$.

3.4. Distribución de probabilidad en la etapa n-ésima

Una vez definida la cadena de Markov en tiempo discreto, se puede obtener la distribución marginal de X_n, es decir, la distribución de la cadena en la etapa n-ésima.

Si se denota la probabilidad de que en la etapa n la cadena de Markov se encuentre en el estado i por $p_i^{(n)} = P(X_n = i)$, para cada etapa se tiene un vector $P^{(n)} = (p_1^{(n)}, \ldots, p_i^{(n)}, \ldots, p_k^{(n)})$ que representa la probabilidad de que la cadena se encuentre en cada uno de los posibles estados en la etapa n. Para poder obtener esta distribución se deberá calcular previamente la matriz de probabilidades de transición en n etapas.

Partiendo de que se conoce la probabilidad de transición en una etapa dada, el siguiente paso es obtener la probabilidad de transición en n etapas,

$$p_{ij}^{(n)} = P(X_{m+n} = j | X_m = i), \qquad \forall n, m \in \mathbb{N}\,; i, j \in E.$$

Éstas se obtienen calculando la potencia n-ésima de la matriz de transición P. Este resultado se obtiene a partir de la ecuación de Chapman-Kolmogorov para cadenas de Markov en tiempo discreto

$$p_{ij}^{(m+n)} = \sum_{g \in E} p_{ig}^{(m)} p_{gj}^{(n)}, \qquad \forall n, m \in \mathbb{N}\,; i, j \in E. \tag{3.8}$$

Intuitivamente, para pasar del estado i al j en $(m + n)$ etapas, se debe pasar por un estado g en m etapas y después ir desde g hasta j en las n etapas restantes. La condición de Markov implica que las dos partes de la transición de i a j son

independientes, por lo que se puede escribir $p_{ij}^{(m+n)}$ como el producto de las probabilidades de transición y por lo tanto $P^{(m+n)} = P^{(m)}P^{(n)}$, donde $P^{(m+n)}$ es la matriz de transición de la etapa $(m+n)$ y tiene como elementos $(p_{ij}^{(m+n)})$.

Tomando $n = 1$ en la ecuación (3.8) se obtiene

$$p_{ij}^{(m+1)} = \sum_{g \in E} p_{ig}^{(m)} p_{gj}, \qquad \forall n, m \in \mathbb{N}\,; i, j \in E. \tag{3.9}$$

Se tiene entonces que $P^{(1)} = P$, $P^{(2)} = P^{(1)}P^{(1)} = P^2$, y así sucesivamente hasta que obtenemos $P^{(n)} = P^{(n-1)}P = P^{(n-2)}P^2 = \ldots = P^{(0)}P^{(n)}$, y con esta fórmula tenemos la distribución de la cadena en la etapa n. Por lo tanto, la probabilidad de transición en n etapas, $p_{ij}^{(n)} = P(X_{m+n} = j | X_m = i)$ se obtiene del elemento (i, j) de la n-ésima potencia de la matriz de transición P; $\forall n, m \in \mathbb{N}\,; i, j \in E$.

3.5. Orden de una Cadena de Markov

El *orden* de una cadena de Markov establece el número de estados anteriores de los cuales depende la probabilidad de un estado, en un instante dado del proceso. Así, dado $E = \{E_1, \ldots, E_k\}$, en una cadena de primer orden tendremos

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n), \tag{3.10}$$

y en una cadena de orden dos

$$\begin{aligned} P(X_{n+1} = i_{n+1} | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i_n) = \\ = P(X_{n+1} = i_{n+1} | X_{n-1} = i_{n-1}, X_n = i_n), \end{aligned} \tag{3.11}$$

para $i \in E$ y $n \in \mathbb{N}$. Análogamente se definiría una cadena de Markov de orden mayor que dos.

3.6. Clasificación de estados

Dentro de los estados de una cadena de Markov en tiempo discreto, se puede hacer una clasificación de éstos según las transiciones permitidas entre ellos y las probabilidades de pasar de un estado a otro. Para realizar dicha clasificación es necesario primero definir los conceptos de probabilidad y tiempos de primera pasada.

Definición 3.3 ***Probabilidad de primera pasada***. *Es la probabilidad de que, empezando en i, la cadena pase por primera vez por el estado j en la etapa n. Se denota por*

$$f_{ij}^{(n)} = P(X_n = j, X_r \neq j, \forall r < n | X_0 = i). \tag{3.12}$$

Definición 3.4 ***Probabilidad de pasada****. Se trata de la probabilidad de que la cadena llegue alguna vez al estado j partiendo del estado i.*

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)} = P(\exists\, n; X_n = j | X_0 = i). \tag{3.13}$$

Definición 3.5 ***Tiempo de primera pasada****. Es el número de etapa en la que la cadena llega por primera vez al estado j cuando parte del estado i. Se determina por la siguiente variable aleatoria:*
$N_{ij} = \{n^o$ de la primera etapa en la cual la cadena está en j partiendo de $i\}$. Se describe entonces

$$f_{ij}^{(n)} = P(N_{ij} = n),\ f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)} = P(N_{ij} < \infty). \tag{3.14}$$

Se establece la siguiente relación entre las probabilidades de transición en n etapas, $p_{ij}^{(n)}$, y las probabilidades de primera pasada ,$f_{ij}^{(n)}$,

$$p_{ij}^{(n)} = f_{ij}^{(1)} p_{jj}^{(n-1)} + f_{ij}^{(2)} p_{jj}^{(n-2)} + \ldots + f_{ij}^{(n-1)} p_{jj} + f_{ij}^{(n)}. \tag{3.15}$$

Se puede hacer entonces la siguiente clasificación del espacio de estados E de una cadena de Markov en tiempo discreto

1. Estado *recurrente*: Un estado $j \in E$ se dice que es recurrente si $f_{jj} = 1$, es decir, es seguro que la cadena va a volver al estado j una vez que ya ha llegado a él en alguna etapa. De otra forma, es recurrente, si y solo si el número de veces que se espera que la cadena pase por él cuando ya parte de él es infinito.

2. Estado *transitorio*: Se dice que un estado $j \in E$ es transitorio si no es recurrente, es decir, $f_{jj} < 1$. En este caso se espera un número finito de visitas a dicho estado.

3. Estado que *comunica* con otro estado: Se dice que un estado $i \in E$ comunica con otro estado $j \in E$, si $f_{ij} > 0$, es decir, existe la posibilidad de que la cadena llegue al estado j partiendo del estado i. Se cumple también si alguna potencia de la matriz de transición otorga probabilidad no nula a la transición entre dos estados. Se dice que una cadena de Markov es *irreducible* si todos sus estados comunican entre sí.

4. Estado que *intercomunican*. Dos estados, $i, j \in E$ intercomunican si i comunica con j y j comunica con i.

5. Estado *efímero*: Se dice que un estado $j \in E$ es efímero si $p_{ij} = 0, \forall i \in E$. No se puede llegar a él desde ningún otro, sólo se puede salir desde él hacia cualquier otro.

6. Estado *absorbente*: Un estado $j \in E$ se dice absorbente si es imposible abandonarlo, es decir, $p_{jj} = 1$. Una vez que se alcanza, la cadena sólo puede mantenerse en él. Además, una cadena de Markov en tiempo discreto se dice que es *absorbente* si tiene al menos un estado absorbente y desde cada uno de los estados es posible alcanzar alguno de los absorbentes en un número finito de etapas. Esta clase de cadenas de Markov se utilizan en Bioinformática para, por ejemplo, encontrar la distancia media entre sucesivas ocurrencias de una "palabra" (como **AAG**) en una pequeña secuencia de ADN.

7. Estado *periódico*: Un estado $i \in E$ se dice que es periódico con período $l > 1$ si l es el menor número tal que todas las secuencias de transiciones que parten del estado i y regresan al estado i tienen una longitud múltiplo de l. Si un estado no es periódico se llama *aperiódico*. Una cadena de Markov se dice *aperiódica* si todos sus estados son aperiódicos.

Asumiremos que todas las cadenas de Markov que veremos en adelante son finitas, aperiódicas e irreducibles.

3.7. Distribuciones estacionarias

Supongamos que una cadena de Markov tiene una matriz de transición P y que en el instante t la probabilidad de que el proceso esté en el estado j es φ_j, $j = 1, 2, \ldots, k$. Esto implica que la probabilidad de que en el instante $t + 1$ el proceso esté en el estado j es $\sum_{r=1}^{k} \varphi_r p_{rj}$. Supongamos que, para todo j, estas dos probabilidades son iguales, por lo tanto

$$\varphi_j = \sum_{r=1}^{k} \varphi_r p_{rj}, \qquad j = 1, 2, ..., k. \tag{3.16}$$

En este caso decimos que la distribución de probabilidad $(\varphi_1, \varphi_2, ..., \varphi_k)$ es *estacionaria*; esto es, no ha cambiado entre los instantes t y $t + 1$, y por lo tanto nunca cambiará.

Definición 3.6 *Una distribución $\Phi = \{\varphi_j\}_{j \in E}$ sobre E se dice estacionaria respecto de una cadena de Markov con matriz de transición P si verifica que $\Phi P = \Phi$. De la misma forma, Φ es una distribución estacionaria si $\sum_{j \in E} \varphi_j = 1$ y se verifica*

$$\varphi_j = \sum_{i \in E} \varphi_i p_{ij}, \qquad \forall j \in E \tag{3.17}$$

Los valores de la distribución estacionaria se pueden interpretar como la proporción final de tiempo que la cadena ha pasado en cada estado a lo largo de su evolución, con probabilidad 1. Se puede decir que es también la proporción, a largo plazo, de etapas en las que la cadena se encuentra en el estado i a lo largo de su evolución, si ha partido de i o de otro estado recurrente que intercomunica con i. Para una cadena de Markov finita, aperiódica e irreducible, existe una única distribución estacionaria.

Si una cadena de Markov es finita, aperiódica e irreducible, entonces si n crece, $P^{(n)}$ se aproxima una matriz $k \times k$ con todas las filas iguales al vector $(\varphi_1, \varphi_2, ..., \varphi_k)$, que es la distribución estacionaria de la cadena de Markov, siendo $P^{(n)}$ la matriz de las probabilidades de transición al cabo de n pasos. La forma de esta matriz muestra que no importa cual sea el estado inicial, o cual sea la distribución de probabilidad del estado inicial, y que la probabilidad de que n pasos después el proceso esté en el estado j se aproxima cada vez más, cuando $n \to \infty$, al valor φ_j.

3.8. Ejemplo: una cadena de Markov como modelo para el ADN

Es improbable que una secuencia aleatoria compuesta por las "letras" A, C, G y T sea un buen modelo para el patrón de nucleótidos en una secuencia génica, como ya vimos en el capítulo anterior. Una cadena de Markov con $\{A, C, G, T\}$ como estados podría ser una mejor aproximación a la realidad: las probabilidades para el nucleótido en la posición $n+1$ dependen del nucleótido en la posición n (sin embargo, en la realidad las dependencias son más complejas). En este caso, se sustituye el concepto del tiempo t por la "posición n" en la secuencia de ADN. Si el espacio de estados es $E = \{A, C, G, T\}$, entonces la matriz de transición será de la forma

$$P = \begin{pmatrix} p_{AA} & p_{AC} & p_{AG} & p_{AT} \\ p_{CA} & p_{CC} & p_{CG} & p_{CT} \\ p_{GA} & p_{GC} & p_{GG} & p_{GT} \\ p_{TA} & p_{TC} & p_{TG} & p_{TT} \end{pmatrix}, \tag{3.18}$$

con $p_{ij} = P(X_{n+1} = j | X_n = i)$, para $n \geq 1$, donde $i, j \in E$.

Además, la distribución estacionaria será (para $k = 4$ estados) $\varphi' = (\varphi_A, \varphi_C, \varphi_G, \varphi_T)$.

3.8.1. Ejemplo

Supongamos que tenemos la secuencia de ADN **CCGAT**, de longitud 5. Vamos a hallar la probabilidad de la secuencia dado el modelo de Markov de primer orden caracterizado por

1. la distribución inicial $P^{(0)} = (p_A^{(0)}, p_C^{(0)}, p_G^{(0)}, p_T^{(0)}) = (0.2, 0.1, 0.1, 0.6)$.

2. la matriz de transición

$$P = \begin{pmatrix} 0.10 & 0.80 & 0.05 & 0.05 \\ 0.35 & 0.10 & 0.10 & 0.45 \\ 0.30 & 0.20 & 0.20 & 0.30 \\ 0.60 & 0.10 & 0.25 & 0.05 \end{pmatrix}. \tag{3.19}$$

Se trata de una cadena de Markov finita, aperiódica e irreducible. El espacio de estados E es en nuestro caso $E = \{A, C, G, T\}$, finito con $k = 4$. La cadena es irreducible ya que todos los estados se comunican entre sí (las probabilidades de transición en cualquier etapa serán no nulas). Además, no existe ningún estado periódico, es decir, no existe l tal que $p_{ii}^{(n)} = 0$ para $n = l, 2l, 3l, \ldots$, o $p_{ii}^{(n)} \neq 0$ para $n = l, 2l, 3l, \ldots$; cualquiera que sea la etapa n.

Sean las variables aleatorias X_n para $n = 0, 1, 2, 3, 4$, que indican el tipo de nucleótido j en la posición n de la secuencia **CCGAT**, $\forall j \in E$. X_0 es una variable aleatoria que determina el nucleótido en la posición inicial de la secuencia. De esta forma, X_1 indica el nucleótido en la primera posición de la secuencia, X_2 el nucleótido en la segunda, y así sucesivamente. Calculemos entonces la probabilidad de la secuencia **CCGAT**, es decir, tenemos que hallar

$$\begin{aligned} P(CCGAT) &= P(X_0 = C) \times P(X_1 = C|X_0 = C) \times P(X_2 = G|X_1 = C) \times \\ &\times P(X_3 = A|X_2 = G) \times P(X_4 = T|X_3 = A) = \\ &= p_C^{(0)} \times p_{CC} \times p_{CG} \times p_{GA} \times p_{AT} \end{aligned}$$

Para empezar, tenemos que $p_C^{(0)} = P(X_0 = C) = 0.1$, teniendo en cuenta la distribución inicial de la cadena.

Por otro lado, $p_{CC} = p_{CC}^{(1)} = P(X_1 = C|X_0 = C) = 0.1$, observando la matriz de transición P indicada anteriormente. De la misma forma, procedemos con todas las probabilidades de transición indicadas en la matriz P, y así obtenemos que la probabilidad de la secuencia de ADN **CCGAT** es, en este caso,

$$\begin{aligned} P(CCGAT) &= p_C^{(0)} \times p_{CC} \times p_{CG} \times p_{GA} \times p_{AT} = \\ &= 0.1 \times 0.1 \times 0.1 \times 0.3 \times 0.05 = 0.000015. \end{aligned}$$

Si considerásemos que esta secuencia se distribuye de forma iid (independiente e idénticamente distribuida), siendo la probabilidad de cada nucleótido la misma que la de la distribución inicial de la cadena de Markov indicada, la probabilidad de la secuencia sería

$$\begin{aligned} P(CCGAT) &= p_C \times p_C \times p_G \times p_A \times p_T = \\ &= 0.1 \times 0.1 \times 0.1 \times 0.2 \times 0.6 = 0.00012, \end{aligned}$$

que es mayor que la probabilidad calculada teniendo en cuenta que la secuencia sigue un modelo de Markov de primer orden como el anterior. Se observa una muy leve diferencia debido al tamaño pequeño de la secuencia.

Capítulo 4

Análisis de las secuencias del ADN

Después de haber introducido la teoría necesaria anterior, pasamos ya al análisis de secuencias de ADN y al estudio sus características.

Desde que el bacteriófago $Phi - X174$ fue secuenciado en 1977-1978, las secuencias de ADN de cientos de organismos han sido decodificadas y guardadas en bases de datos. Esos datos son analizados para determinar, entre otras cosas, los genes que codifican ciertas proteínas. Una comparación de genes en una especie o entre especies puede mostrar similitudes entre funciones de proteínas, o relaciones entre especies. Con la creciente cantidad de datos, desde hace mucho se ha vuelto poco práctico analizar secuencias de ADN manualmente. Hoy en día, se utilizan programas informáticos para estudiar el genoma de miles de organismos, conteniendo miles de millones de nucleótidos. Estos programas pueden compensar mutaciones (con bases intercambiadas, borradas o insertadas) en la secuencia de ADN, para identificar secuencias que están relacionadas, pero que no son idénticas. Una variante de este alineamiento de secuencias se usa en el "proceso de secuenciación".

El método de secuenciación conocido como "shotgun" o "por perdigonada" fue utilizado por el Instituto de Investigación Genómica para secuenciar el primer genoma de una bacteria, la *Haemophilus influenzae*. No da una lista secuencial de nucleótidos, pero en cambio nos ofrece las secuencias de miles de pequeños fragmentos de ADN (cada uno de aproximadamente 600 a 800 nucleótidos de largo). Las terminaciones de estos fragmentos se superponen y, cuando son alineados de la manera correcta, constituyen el genoma completo del organismo en cuestión. El secuenciamiento "shotgun" proporciona datos de secuencia rápidamente, pero la tarea de ensamblar los fragmentos puede ser bastante complicada para genomas muy grandes. En el caso del Proyecto del Genoma Humano, se requirieron varios meses de tiempo de procesador para ensamblar los fragmentos. El "shotgun sequencing" es el método elegido para todos los genomas secuenciados hoy en día, y los algoritmos de ensamblado genómico son un área crítica de la investigación en Bioinformática. No profundizaremos en el análisis de este método debido a su complejidad técnica ([40]).

Otro aspecto importante de la Bioinformática en el análisis de secuencias es la búsqueda automática de genes. Veremos de forma introductoria cómo modelizar secuencias de genes en el capítulo 5, como aplicación de los Modelos de Markov Ocultos.

En su nivel más elemental, como ya comentamos, la estructura del ADN puede pensarse como largas secuencias de nucleótidos. Estas secuencias están organizadas en secuencias codificantes, o genes, separadas por largas regiones intergénicas de secuencias no codificantes. La mayoría de los genes eucariotas tienen un nivel adicional de organización: dentro de cada gen, las secuencias codificantes (*exones*) son normalmente interrumpidas por tramos de secuencias no codificantes (*intrones*). Durante la transcripción del ADN en ARN mensajero (ARNm), los intrones se eliminan y no aparecen en el último ARNm, el cual se traduce en secuencia de proteína. Las regiones intergénicas y los intrones tienen diferentes propiedades estadísticas que los exones. Para capturar estas propiedades en un modelo, podemos construir procedimientos estadísticos y comprobar si una parte no caracterizada de ADN es parte de la región codificante del gen. El modelo está basado en un conjunto de datos "training" o datos "de entrenamiento" tomados de secuencias ya conocidas. En el modelo más sencillo, se asume que los nucleótidos en posiciones distintas son independientes y están idénticamente distribuidos (iid). Si éste es el caso, las diferencias entre ADN codificante y no codificante podrían detectarse por las diferencias entre las frecuencias de los cuatro nucleótidos en los dos casos distintos (intrón vs exón). Estas distribuciones pueden estimarse mediante los datos de entrenamiento. Si no se utilizan datos de entrenamiento como referencia, el análisis es más complicado. En las aplicaciones prácticas que presentaremos en esta memoria sí los utilizaremos.

4.1. Contrastes de independencia

La precisión de los procedimientos a llevar a cabo depende de la precisión de las suposiciones hechas. Podemos intuir que suponer que hay independencia entre los nucleótidos suele ser una simplificación excesiva, puesto que puede haber correlaciones, por ejemplo, entre los nucleótidos debido a su pertenencia a uno u otro codón. Por ello, es importante, entre otras cosas, desarrollar un contraste de independencia sobre la secuencia de nucleótidos. Este contraste está basado en el análisis de una cadena de Markov. Al utilizar cadenas de Markov en el análisis de secuencias, el concepto del "tiempo t" se reemplaza por la "posición a en la secuencia". Nuestras variables aleatorias o sucesos serán "los nucleótidos en posiciones dadas de la secuencia de ADN" y, por tanto, el espacio de estados estará compuesto por los cuatro nucleótidos: $E = \{A, C, G, T\}$, de tamaño $k = 4$. Desarrollamos entonces un test de independencia chi-cuadrado χ^2 sobre la secuencia de nucleótidos que queremos analizar, que contrasta la hipótesis nula de que los nucleótidos en posiciones distintas sean independientes, frente a la hipótesis alternativa de que un nucleótido en una posición dada dependa del nucleótido en la posición anterior. De esta forma, la

distribución del contraste es una cadena de Markov en la cual todas las filas de la matriz de transición son idénticas. Por lo tanto, la hipótesis alternativa la pensamos como una cadena de Markov donde la probabilidad de un nucleótido en una posición dada, dependa del nucleótido en la posición anterior, es decir, una cadena de Markov de orden uno. Así, como ya vimos en el ejemplo del capítulo anterior, bajo la hipótesis nula de independencia entre nucleótidos, y dada la secuencia de ADN $X =$ **GATTACA**, por ejemplo, la probabilidad de tal secuencia bajo el modelo de independencia será $P(X) = p_G p_A p_T p_T p_A p_C p_A = p_A^3 p_C^1 p_G^1 p_T^2$, donde p_i indica la probabilidad del nucleótido i en la secuencia, $\forall i \in \{A, C, G, T\}$.

Este contraste es de interés para evaluar si el modelo de Markov bajo la hipótesis alternativa describe la realidad significativamente mejor que el modelo de independencia y, por tanto, podría aumentar la precisión de nuestros procedimientos de predicción. Además, si lo hace, podría considerarse incluso el modelo con un nivel de dependencia más complejo (una cadena de Markov con mayor orden).

Supongamos que la longitud de la secuencia de ADN que queremos analizar (nuestra muestra) es n. El contraste estadístico de independencia es un *contraste de asociación* en una tabla 4×4 de doble entrada como la siguiente, denominada *tabla de contingencia*.

	A	C	G	T	Total
A	n_{11}	n_{12}	n_{13}	n_{14}	$n_{1,}$
C	n_{21}	n_{22}	n_{23}	n_{24}	$n_{2,}$
G	n_{31}	n_{32}	n_{33}	n_{34}	$n_{3,}$
T	n_{41}	n_{42}	n_{43}	n_{44}	$n_{4,}$
Total	$n_{,1}$	$n_{,2}$	$n_{,3}$	$n_{,4}$	n

Tabla 4.1: Tabla de contingencia de los cuatro nucleótidos.

Tenemos entonces un contraste de independencia con la hipótesis nula

$$H_0 : p_{ij} = p_i \times p_j, \qquad \forall i, j \in E.$$

Los datos deben aparecer como se muestra en la tabla 4.1, donde las filas representan el nucleótido en la posición a y las columnas el nucleótido en la posición $a+1$. El elemento (i, j) de la tabla representa la frecuencia observada n_{ij}, que indica el número de transiciones del estado o nucleótido i al estado o nucleótido j en una etapa, para $i, j \in E$; y $\sum_{i,j \in E} n_{ij} = n$. Así, cada uno de los elementos $n_{i,}$ y $n_{,j}$ representan la suma de la fila i (número de transiciones desde el estado i) y la columna j (número de transiciones que llegan al estado j), respectivamente, $\forall i, j \in E$. La idea es realizar el contraste anterior comparando las frecuencias esperadas bajo la hipótesis nula, $T_{ij} = np_ip_j$, con las observadas, $O_{ij} = n_{ij}$. Si las cantidades p_i y p_j

no son conocidas, han de ser estimadas a partir de las frecuencias observadas de la siguiente forma

$$\widehat{p}_i = \frac{n_{i,}}{n}, \qquad \widehat{p}_j = \frac{n_{,j}}{n}, \tag{4.1}$$

y por lo tanto

$$T_{ij} = n\widehat{p}_i\widehat{p}_j = \frac{n_{i,}n_{,j}}{n}, \tag{4.2}$$

lo que hace perder $(k-1)+(k-1)$, con $k=4$, grados de libertad adicionales al estadístico de contraste

$$X^2_{(k-1)+(k-1)} = \sum_{i=1}^{k}\sum_{j=1}^{k}\frac{(n_{ij}-T_{ij})^2}{T_{ij}} \simeq \chi^2_{(k-1)+(k-1)}. \tag{4.3}$$

que sigue, bajo la hipótesis nula, una distribución aproximada chi-cuadrado con tales grados de libertad. De esta forma, rechazaremos H_0 si

$$X^2_{(k-1)+(k-1)} > \chi^2_{(k-1)+(k-1),1-\alpha},$$

y el p-valor correspondiente será menor o igual que α, para $\alpha = 0.05$. Por lo tanto, la hipótesis nula de independencia resulta ser la hipótesis nula de no asociación en la tabla.

Pruebas de este tipo muestran que nucleótidos contiguos en posiciones de secuencias de ADN son a veces dependientes, y el modelo de una cadena de Markov de primer orden se ajusta a los datos reales significativamente mejor que el modelo de independencia, tanto en regiones de intrones como de exones. Modelos homogéneos de Markov de orden mayor y modelos no homogéneos a veces pueden ajustarse mejor a los datos.

Podemos ahora extender el análisis de independencia realizado a los contrastes de cadenas de Markov de orden superior (mayor que 1), por un procedimiento de razón de verosimilitud generalizada. Presentamos este análisis en términos de secuencias de ADN también: queremos saber cuál es el orden de la cadena de Markov que describe significativamente mejor las secuencias de ADN.

La estructura probabilística de los nucleótidos en una secuencia de ADN se describe por una cadena de Markov de orden $d \geq 1$ si la probabilidad de que cualquier nucleótido ocurra en una posición dada depende de los nucleótidos de las d anteriores posiciones. Las probabilidades de transición para el caso $d=3$, por ejemplo, son de la forma

$$P(X_{n+3} = i_{n+3} | X_n = i_n, X_{n+1} = i_{n+1}, X_{n+2} = i_{n+2}),$$

donde X_a denota el nucleótido en la posición a del la secuencia, y los i_a son tipos específicos de nucleótidos, $\forall i \in E$ y $a = 1, 2, \ldots, n$. Esta notación muestra que para un d general, habrá 3×4^d probabilidades de transición entre distintos nucleótidos.

Supongamos que la hipótesis nula es que la cadena de Markov es de orden $d-1$, y la hipótesis alternativa que es de orden d, para algún valor de d (lo discutido en la sección anterior es análogo con $d=1$). En términos del contraste anterior,

$$\begin{cases} H_0\text{: el nucleótido en la posición } a \text{ es independiente del de la posición } a-d. \\ H_a\text{: el nucleótido en la posición } a \text{ depende del de la posición } a-b. \end{cases}$$

para todo $b = 1, \ldots, d$.

Aplicamos el test de razón de verosimilitud generalizada ([8]), según el cual

Teorema 4.1 *Sea $X_1, \ldots, X_n$ una muestra aleatoria de una población con función de densidad de probabilidad multiparamétrica dada por $f(x, \theta)$, con $\theta = (\theta_1, \ldots, \theta_m)$. Si $\theta \in \Theta_0$, en que Θ_0 es el correspondiente espacio paramétrico bajo H_0. Entonces el estadístico $T(X) = -2log(\lambda)$ tiene una distribución asintótica χ^2_{m-r}, donde r es el número de parámetros de θ que ha sido completamente especificado en la hipótesis nula y donde λ tiene la siguiente expresión*

$$\lambda = \frac{sup_{\theta \in \Theta_0} f_n(x, \theta)}{sup_{\theta \in \Theta} f_n(x, \theta)} = \frac{supL_0}{supL} = \frac{f_n(x, \widehat{\theta}_0)}{f_n(x, \widehat{\theta})}. \tag{4.4}$$

En nuestro caso, la función de densidad de probabilidad corresponde a la matriz de transición de la cadena de Markov, $\widehat{P}$, donde los elementos de θ representan cada una de las probabilidades de transición estimadas entre nucleótidos, $\widehat{p}_{ij} = \frac{n_{ij}}{n_{i,}}$, $\forall i, j \in E$. El número $n_{i,}$ representa la suma de todos los procesos que salen del estado i considerado en cada caso, es decir $n_{i,} = \sum_{j \in E} n_{ij}$. Así, la suma por filas de la matriz $\widehat{P}$ es lógicamente 1. Por lo tanto, la matriz de transición $\widehat{P}$ estimada por máxima verosimilitud se puede obtener a partir de la tabla de contingencia de las variables estado anterior y estado actual (tabla 4.1), calculando las distribuciones marginales por filas. El número de parámetros r de θ bajo la hipótesis nula es $3 \times 4^{(d-1)}$, por definición de orden de una cadena de Markov. Por tanto, el número de parámetros (m) bajo la hipótesis alternativa es 3×4^d. Teniendo en cuenta esto, el estadístico T de la definición anterior tendrá una distribución asintótica chi-cuadrado con $m - r = 3 \times 4^d - 3 \times 4^{(d-1)} = 9 \times 4^{(d-1)}$ grados de libertad.

En este caso, la expresión de λ será la siguiente

$$\lambda = \frac{\prod_{i,j \in E} \widehat{p}_{ij0}{}^{n_{ij}}}{\prod_{i,j \in E} \widehat{p}_{ij}^{n_{ij}}}, \tag{4.5}$$

donde $\widehat{p}_{ij0}$ denotan las probabilidades de transición en una cadena de Markov de orden $d-1$, es decir, bajo la hipótesis nula.

En general, se conoce que incluso secuencias de ADN de regiones intergénicas suelen seguir modelos de cadenas de Markov de orden superior a uno.

4.2. Contraste de bondad de ajuste

Si el contraste de independencia no es aceptado, se aplica otro test de interés para saber si la distribución estacionaria de la cadena de Markov es uniforme. De esta forma, si fuese así, los cálculos llevados a cabo con los datos de entrenamiento para hallar los parámetros del modelo se facilitarían bastante. Aplicaremos entonces un contraste de bondad de ajuste a tal distribución. La hipótesis nula será que las distribuciones de probabilidad estacionaria y uniforme son idénticas, y la hipótesis alternativa que ambas distribuciones son diferentes. Podemos discutirlo en el caso de una cadena de Markov de primer orden, ya que para cadenas de órdenes superiores es análogo.

Si denotamos por Φ la distribución estacionaria y por U a la función de distribución uniforme, el contraste de hipótesis planteado se puede escribir de la forma

$$\begin{cases} H_0 : \Phi \cong U. \\ H_a : \Phi \ncong U. \end{cases}$$

En el caso de primer orden, la ecuación (3.17) muestra que una condición necesaria y suficiente para que la distribución estacionaria sea uniforme es que las probabilidades de transición en cada columna de la matriz de transición sumen 1. Es el siguiente teorema.

Teorema 4.2 *Si la matriz de transición P de una cadena de Markov con k estados es doblemente estocástica, entonces la distribución uniforme $U(i) = \frac{1}{k}$ para todo $i \in E$, es una distribución estacionaria.*

Demostración 4.2.1 *Observamos que*

$$\sum_{i \in E} U(i) p_{ij} = \frac{1}{k} \sum_{i \in E} p_{ij} = \frac{1}{k}, \qquad \forall j \in E \tag{4.6}$$

de modo que la distribución uniforme satisface la condición $UP = U$ que define una distribución estacionaria (junto con que, por ser distribución de probabilidad, $\sum_{i \in E} U(i) = 1$). Vemos además, que si la distribución estacionaria es uniforme, necesariamente la matriz P es doblemente estocástica, es decir, la suma por filas y columnas de P ha de ser 1.

En nuestro caso, suponemos entonces que la distribución uniforme es $U = \frac{1}{4}$. Esto implica que, por ejemplo, los elementos en la cuarta fila de la matriz de transición están determinados por los elementos de las primeras tres filas. La probabilidad de cualquier secuencia observada de ADN puede calcularse bajo la hipótesis nula de que la distribución estacionaria es uniforme. Ya que bajo esta hipótesis los elementos de cada fila y cada columna de la matriz de transición deben sumar 1, hay 9 parámetros libres (4×3 había cuando solo las filas sumaban 1, menos 3 que se restan cuando

también las columnas lo hacen). La probabilidad de la secuencia observada puede maximizarse con respecto a estos parámetros. Bajo la hipótesis alternativa, la única restricción es que los elementos de cualquier fila de la matriz de transición deben sumar 1, y por lo tanto habrá 12 parámetros libres. Así, de la misma forma, la probabilidad de la secuencia observada puede ser maximizada respecto a estos parámetros.

De estas dos probabilidades puede calcularse un estadístico de verosimilitud $-2log\lambda$ como el señalado en la sección anterior, donde λ es

$$\lambda = \frac{sup_{\theta\in\Theta_0} f_n(x,\theta)}{sup_{\theta\in\Theta} f_n(x,\theta)} = \frac{supL_0}{supL} = \frac{f_n(x,\widehat{\theta}_0)}{f_n(x,\widehat{\theta})} = \frac{\frac{1}{4}}{sup_E\Phi}. \tag{4.7}$$

Según este contaste, bajo la hipótesis nula, este estadístico tiene una distribución asintótica chi-cuadrado con $m - r = 12 - 9 = 3$ grados de libertad.

De esta forma, podemos aplicar un contraste de bondad de ajuste entre ambas distribuciones y comprobar si la distribución estacionaria se identifica significativamente con la distribución uniforme, lo que facilita los cálculos a la hora de modelizar secuencias de ADN.

4.3. Modelización de "señales" en el ADN

En el contexto de la genómica, la *anotación* o *puntuación* es el proceso de marcado de los genes y otras características biológicas de la secuencia de ADN. El primer sistema software de anotación de genomas fue diseñado en 1995 por Owen White, director de Bioinformática de la escuela de medicina de la universidad de Maryland, quien fue miembro del equipo que secuenció y analizó el primer genoma de un organismo independiente en ser decodificado, la bacteria *Haemophilus influenzae* (responsable de un amplio rango de enfermedades como meningitis, epiglotitis, neumonía, sepsis y otras de menor gravedad). White construyó un software para localizar los genes, el ARN de transferencia, y otras características; así como para realizar las primeras atribuciones de función a esos genes. La mayoría de los actuales sistemas de anotación genómica trabajan de forma similar, pero los programas disponibles para el análisis del genoma se encuentran en continuo cambio y mejora.

En este contexto, se conoce que los genes contienen "señales" en el ADN, que son secuencias de ADN con un propósito específico: indicar, por ejemplo, el comienzo y final de la región transcrita, los límites de los exones e intrones, así como otras características. La maquinaria de la célula utiliza estas señales para reconocer el gen, para editarlo correctamente, y para traducirlo apropiadamente en proteína. Si la naturaleza fuese bondadosa, cada señal consistiría en una única secuencia de ADN que no aparecería en ningún otro lugar del ADN excepto donde sirviese para su propósito específico. En la realidad, hay muchas secuencias de ADN que realizan la misma función de señal; llamamos a éstas "miembros" de una señal. Además, los miembros

de las señales también aparecen aleatoriamente en el ADN no funcional, haciendo difícil clasificar las señales funcionales de las no funcionales. En la práctica, no todos los miembros de una señal son conocidos. Nuestro objetivo es utilizar los miembros conocidos para evaluar la probabilidad de que una nueva secuencia no caracterizada de ADN sea también un miembro de la señal. Supondremos que los diferentes miembros surgen de antepasados comunes mediante procesos estocásticos, por lo tanto, es razonable construir un modelo estadístico de los datos. Algunas señales requieren solamente modelos simples, mientras que otras necesitan modelos más complejos. Cuando el modelo requerido es complejo y los datos son limitados, debemos tener cuidado eligiendo suposiciones simplificadoras en el modelo para utilizar los datos de la manera más eficiente posible.

Asumimos que todos los miembros de la señal de interés tienen la misma longitud, que denotaremos por n. Esta suposición no es demasiado restrictiva, ya que los miembros de muchas señales tienen la misma longitud, y para los que no la tengan, podemos capturar porciones de ellos, que es normalmente suficiente. Para modelar las propiedades de cualquier señal debemos tener un conjunto de datos de entrenamiento, esto es, una gran cantidad de datos en la cual los miembros de las señales sean conocidos.

Consideraremos ahora algunos de los modelos de señales básicos que se utilizan en Bioinformática.

4.3.1. Matrices de Peso. Independencia

El contraste de hipótesis sobre si una secuencia no caracterizada de ADN es miembro de una señal dada, es más fácil de llevar a cabo cuando el nucleótido en cualquier posición de la señal es independiente de los nucleótidos en cualquier otra posición de esa señal. Es, por lo tanto, necesario realizar un contraste de independencia sobre si el nucleótido en la posición a en una señal es independiente del nucleótido en la posición b, sin necesidad de que a y b sean contiguos. Esto puede hacerse de varias formas. Es natural generalizar el contraste de independencia descrito en la tabla 4.1, que comprueba la independencia en posiciones contiguas. En esta generalización, "posición a" es reemplazada por "posición a en la señal" y "posición $a+1$" se reemplaza por "posición b en la señal". Así, n_{ij} es interpretado como el número de veces que, en los datos, el nucleótido i ocurre en la posición a de la señal y el nucleótido j ocurre en la posición b de la señal. Este contraste se utiliza luego para todos los pares a y b; con $i, j \in E$ y $a, b = 1, \ldots, n$.

Por tanto, nos planteamos un contraste de independencia chi-cuadrado entre las posiciones de los nucleótidos en la señal, es decir la hipótesis nula es

H_0: el nucleótido en la posición a es independiente del de la posición b,

$\forall\, a, b = 1, \ldots, n$. El contraste se desarrolla análogamente al test de independen-

cia para cadenas de Markov de orden uno, expuesto en el presente capítulo (donde las posiciones en la secuencia sí eran contiguas).

Supongamos que, como resultado de este contraste, puede asumirse independencia. Se construye entonces una matriz $4 \times n$, donde cada fila corresponde a uno de los 4 nucleótidos y cada columna a las posibles posiciones de la señal de longitud n. Su posición o entrada (i, a) es la proporción de casos en los datos que el nucleótido i ocurre en la posición a de la señal. Nos referimos a ésta como una *matriz de peso* y la denotaremos por M.

Un ejemplo para el caso $n = 5$ se presenta en la siguiente matriz

$$M = \begin{pmatrix} 0.33 & 0.34 & 0.19 & 0.20 & 0.21 \\ 0.31 & 0.18 & 0.34 & 0.30 & 0.25 \\ 0.22 & 0.27 & 0.23 & 0.24 & 0.21 \\ 0.14 & 0.21 & 0.24 & 0.26 & 0.33 \end{pmatrix}. \tag{4.8}$$

Los elementos de la columna a de la matriz dan (de forma aproximada) las probabilidades para los cuatro nucleótidos (A, G, C y T, respectivamente) en la posición a de la señal, $a = 1, \ldots, n$. La matriz M define la probabilidad $P(X|M)$ para cualquier secuencia (señal) X de longitud n.

Las matrices de peso se utilizan como una componente de un algoritmo para búsqueda de genes que veremos brevemente en el siguiente capítulo. En general, al modelizar un gen, se necesitan matrices de peso para modelizar ciertas "regiones" de éste ([40]).

4.3.2. Dependencias de Markov

Si los nucleótidos en las posiciones de la señal no son independientes, una posibilidad es que sus posiciones a lo largo de la señal sigan un modelo de Markov de primer orden. Bajo estas hipótesis, los nucleótidos en cada posición tienen una distribución de probabilidad dependiente del nucleótido en la posición anterior. Así, estas probabilidades se disponen una matriz de transición 4×4 cuyo elemento (i, j) es la probabilidad de que el nucleótido j esté en la posición a, dado el nucleótido i en la posición $a - 1$. Los elementos de esta matriz son estimados a partir de los datos, como ya hemos visto en el contraste de independencia presentado para secuencias de ADN, no necesariamente señales.

En general, podrían considerarse dependencias en cadenas Markov de mayor orden. Sin embargo, estas matrices de transición pueden ser muy grandes a medida que el orden de la cadena de Markov crece y, por lo tanto, la cantidad de datos que se necesitan para satisfacer la estimación puede ser excesiva. Por ello, cuando el conjunto de los datos es limitado, debemos economizar y capturar solamente las dependencias "más informativas" en nuestro modelo, que serán aquellas posiciones

donde encontremos valores mayores de una tabla de contingencia como la tabla 4.1. Un método para hacer esto lo vemos a continuación.

Descomposición de Dependencia Máxima

Puede ser imposible, debido a los datos limitados, obtener satisfactoriamente estimaciones de las distribuciones de probabilidad de una señal dada, en los casos de cadenas de Markov de orden mayor que uno. Esto motiva la búsqueda de un método de capture aquellas dependencias que sean "más informativas". Describiremos ahora la aproximación, conocida como *descomposición de dependencia máxima* ([9]), para solucionar este problema.

Supongamos que deseamos modelizar una señal de longitud n. El primer paso es encontrar una posición que tenga la "mayor influencia" en las otras. Para ello, construimos una tabla $n \times n$ cuyas entradas (a, b) sean los valores del estadístico chi-cuadrado obtenidos de una tabla de contingencia como tabla 4.1, pero que compare los nucleótidos de una posición fija a con los de otra posición b (en lugar de $a+1$). Si la hipótesis de independencia es cierta, esperamos encontrar alrededor de un valor significativo sobre 20, con un error de Tipo I o α del 5 %. Así, si algunos de los valores del estadístico chi-cuadrado observados son "un poco" significativos, podríamos concluir que las frecuencias observadas en la muestra no difieren significativamente de las que teóricamente deberían haberse obtenido bajo independencia, y por tanto no rechazaríamos la hipótesis nula. Si "bastantes" valores son significativos, o si existen valores con "grandes" niveles de significación, entonces podríamos concluir que los nucleótidos en varias posiciones no son independientes, y rechazaríamos la hipótesis nula al haber encontrado evidencia en el sentido de que las dos variables consideradas están relacionadas: nucleótido en posición a con nucleótido en posición b.

La tabla siguiente proporciona un ejemplo, con $n = 5$, en el cual muchos de los valores chi-cuadrado son significativos al nivel del 5 % (indicados por asteriscos). La fila con la suma mayor da una indicación de cuál es la posición que tiene mayor influencia sobre las otras $n - 1$ posiciones restantes.

	1	2	3	4	5	total
1	0	34.2*	7.1	37.2*	2.8	81.3
2	34.2*	0	0.4	72.4*	4.5	111.5
3	7.1	0.4	0	15.3	98.3*	121.1
4	37.2*	72.4*	15.3	0	14.2	139.1
5	2.8	4.5	98.3*	14.2	0	119.8

Tabla 4.2: Ejemplo.

En este ejemplo, la posición 4 tiene el mayor total por filas, por lo tanto la toma-

remos como la posición de mayor influencia y trataremos de evaluar esta influencia. Es este caso decimos que *dividimos en la posición 4*.

Construimos primero un modelo que determine las distribuciones para las posiciones 1, 2, 3 y 5 condicionales de la posición 4. Para ello, dividimos la secuencia dada en 4 conjuntos de secuencias, cada uno determinado por el nucleótido en la posición 4. Por lo tanto, para cada nucleótido x tendremos un conjunto T_x constituido por aquellas secuencias donde haya una x en la posición 4. Después calculamos $p_x = \frac{n_x}{d}$, para cada $x = A, G, C, T$, donde n_x es el número de miembros de los datos que tienen el nucleótido x en la posición 4, y $d = \sum n_x$. Luego, para cada $x = A, G, C, T$, calculamos una matriz de peso 4×4, M_x, de las secuencias T_x, para las posiciones 1, 2, 3 y 5, de la misma forma que la calculada en el caso de independencia entre nucleótidos de la señal.

El modelo de dependencia tiene entonces la distribución $\{p_A, p_G, p_C, p_T\}$ junto con las cuatro matrices de peso $\{M_A, M_G, M_C, M_T\}$. Así, para cualquier secuencia X de longitud 5 calcularemos la $P(X)$ como sigue: si el nucleótido x ocurre en la posición 4 de X, la matriz de peso M_x se utiliza para asignar una probabilidad p_a a las posiciones $a = 1, 2, 3$ y 5 de X. Entonces, $P(X) = p_x \times p_1 \times p_2 \times p_3 \times p_5$.

En general, algunos o todos los conjuntos T_x serán suficientemente grandes (secuencias de longitud grande) para que podamos repetir el proceso entero recursivamente, dividiendo T_x en una de las posiciones 1, 2, 3 y 5. Es este caso, para cada T_x suficientemente grande construiremos una tabla similar a la 4.2, esta vez 4×4, para encontrar la posición de esas cuatro que tenga la influencia mayor en las otras tres. Entonces, descompondremos T_x en T_{xy}, para $y = A, G, C$ y T; continuaremos a T_{xyz} y así sucesivamente, siempre y cuando haya datos suficientes. Una regla que podemos utilizar es parar de buscar posiciones influyentes cuando el conjunto $T_{xyz...}$ tenga menos de 100 secuencias. Cuando ese límite se alcanza, las posiciones restantes se modelan con una matriz de peso, con la que se procede de la forma explicada en la sección anterior.

4.4. Análisis de Patrones

Un patrón es una secuencia de ADN que se repite en determinadas posiciones de otra secuencia más larga, correspondiente a distintas partes de la estructura del ADN.

Supongamos que estamos interesados en una "palabra", por ejemplo **GAGA**. Nos preguntamos las dos siguientes cuestiones de una secuencia de ADN iid de longitud n: ¿cuál es el número medio de veces que esta palabra ocurre en un segmento de longitud n? y ¿cuál es la longitud media entre una ocurrencia de esta palabra y la siguiente?.

Este análisis asume que los tipos de nucleótidos en diferentes posiciones son independientes. La suposición de independencia se utiliza para poder introducir algunas características de las propiedades de los patrones en un marco simple. Como existen dependencias entre nucleótidos en posiciones contiguas, el análisis general se extiende al caso de la dependencia de una cadena de Markov.

Hay varias razones por las cuales debemos preguntarnos estas cuestiones. Una de ellas es que puede haber alguna razón "a priori" para sospechar que la palabra **GAGA** ocurre significativamente más a menudo de lo que debería si los nucleótidos son generados en una manera iid. Para comprobar esto, es necesario tener en cuenta aspectos probabilísticos de la frecuencia de esta palabra asumiendo iid. Quizás sorprendentemente, las frecuencias de otras palabras como **GGGG**, **GAAG**, y **GAGC** son a menudo diferentes de los de la palabra **GAGA**, incluso bajo iid. Una segunda razón ha sido discutida por Bussemaker et al. ([26]). Aquí, el objetivo es descubrir señales promotoras buscando patrones comunes de ADN sobre regiones de genes. Esto se puede hacer creando un diccionario de palabras de longitudes diferentes, cada una con una asignación de probabilidad. Cualquier palabra que ocurra más frecuentemente que la esperada en determinadas regiones de genes, es una candidata para tal señal. El análisis llega a ser complicado, ya que no hay una palabra de interés "a priori", y, en efecto, tampoco tenemos una longitud de palabra que pueda definirse por adelantado y buscarse específicamente.

Para más información sobre el análisis de patrones podemos consultar [36], donde se estudia una fórmula de detección de secuencias patrones calculando sus frecuencias esperadas.

4.5. Búsqueda de repeticiones en secuencias de ADN

Describimos ahora las posibles repeticiones de secuencias de ADN presentes en un genoma eucariota y su clasificación de acuerdo con su longitud y forma. También abordaremos el problema computacional para detectarlas y los enfoques utilizados. Para más información y detalles véase [37].

En un genoma eucariota, los nucleótidos no se encuentran en igual cantidad, A y T se encuentran en un 60 % (30 % cada uno), y C, G en un 40 % (20 % cada uno). Pero un genoma no sólo varía en la cantidad de nucleótidos que contiene, también presenta una distribución no uniforme de los nucleótidos a lo largo de la secuencia genómica. Incluso existen algunos segmentos con una alta concentración de un par de nucleótidos (A, T) o (C, G) denominados *isocoros*. Estos isocoros son objeto de estudio ya que presentan una alta concentración de secuencias codificantes. Oliver et al. ([27]) propusieron un método para detectar segmentos (isocoros o dominios que

presenten una composición similar) denominado *segmentación entrópica*. Además de los isocoros, es interesante estudiar repeticiones de secuencias en un genoma ya que, principalmente los organismos eucariotas, contienen un alto número de secuencias repetidas (de longitud, composición y frecuencia variables). Estas repeticiones se pueden encontrar en los genomas de dos formas: "en tandem", dos o más copias consecutivas de un patrón, o dispersas aleatoriamente a lo largo del genoma.

4.5.1. Repeticiones en tandem

Estas repeticiones no están definidas claramente todavía, no obstante se pueden clasificar de la siguiente forma

Clase	Long - Nucleótidos	Frec - miles
Satélites	superior a 100	1000
Minisatélites	9 - 100	100
Microsatélites	1 - 8	10 - 1000

Tabla 4.3: Repeticiones en tandem.

Los microsatélites son muy variables, muy abundantes y están distribuidos por todo el genoma. Se utilizan principalmente para pruebas de paternidad y construcción de mapas genéticos.

4.5.2. Repeticiones dispersas

Las repeticiones dispersas o intercaladas se clasifican de acuerdo con su longitud en cortas, denominadas SINE (*short interspersed repeat*) y en largas, denominadas LINE (*long interspersed*). Estas repeticiones pueden estar formadas por la combinación de nucleótidos en diferente orden y cantidad.

Las repeticiones SINE son fragmentos de ADN cortos repetidos millones de veces y dispersos por todo el genoma. En el genoma humano, se estima que tienen una longitud de 100 a 300 pares de bases y se repiten 1,5 millones de veces (13 % del genoma humano). Las repeticiones LINE son fragmentos de ADN de gran tamaño repetidos miles de veces y dispersos por todo el genoma. En el genoma humano se estima que tienen una longitud de 6,000 a 8,000 pares de bases y se repiten 85 mil veces (21 % del genoma humano).

4.5.3. Estudio de repeticiones

La automatización en la obtención de secuencias biológicas ha generado un gran número de estructuras primarias de genomas y proteínas, las cuales demandan de herramientas computacionales para su correcta y eficiente anotación. La importancia del estudio de patrones repetidos aparece desde el proceso de secuenciación.

Durante este proceso, se detectó la presencia de bloques de ADN repetidos, lo cual dificultó el ensamblado del genoma. El proceso de secuenciación automático divide el genoma en miles de fragmentos que luego se amplifican y por último se secuencian ensamblándolos por superposición de sus extremos. Pero antes de unir dos fragmentos, el algoritmo debe determinar si los fragmentos en cuestión son producto de la "clonación" (son el mismo fragmento) o se trata de un fragmento repetido en el genoma. Es el procedimiento "shotgun" que ya comentamos al inicio de este capítulo.

En el genoma del hombre, al igual que en el de muchos organimos, se presentan secuencias de nucleótidos repetidos (patrones o secuencias "motif"). Estos patrones generalmente tienen alguna relación funcional o estructural, ya sea en la secuencia de ADN o en la proteína, y se cree que por esta razón la evolución los ha preservado. Los patrones repetidos pueden ser útiles en tareas como: validación de métodos de clasificación de proteínas, asociación directa con alguna función y/o estructura de una proteína, predicción o identificación de dominios funcionales, o predicción de la estructura tridimensional. Específicamente, para el genoma humano, se estima que el 10 % está formando por secuencias repetidas en tandem. Estas repeticiones son probablemente resultado de una replicación inexacta del ADN que con el tiempo se fijaron en el genoma. El estudio de repeticiones en tandem tiene aplicación directa en medicina debido a que algunos patrones repetidos han sido asociados a enfermedades.

La investigación en el descubrimiento y caracterización de motifs para genomas y proteínas, ha producido una serie de herramientas informáticas, así como la creación de bases de datos bioinformáticas para administrar la información generada. Entre las principales bases de datos se encuentran, por ejemplo: BLOCKS, STRBase (sobre repeticiones cortas en tandem), PROSITE (base de datos de familias de proteínas), Pfam, etc.

4.5.4. El problema de la detección de patrones

La estructura primaria de un genoma se representa computacionalmente como un conjunto de archivos de texto (un archivo para cada cromosoma). Estos archivos contienen la secuencia de nucleótidos. Cada nucleótido se representa con una letra mayúscula A, C, G y T, y si aún no se ha determinado el nucléotido en alguna posición del genoma se utiliza la letra N. El número de repeticiones a buscar se incrementa exponencialmente con la longitud del patrón de interés. En la tabla inferior se ilustra el número total de combinaciones posibles de un patrón de longitud 1 hasta 20.

Para un patrón de longitud n, existen 4^n combinaciones posibles, razón por la cual este problema presenta un reto computacional de interés. La identificación de patrones repetidos en un genoma puede estar dirigida a la búsquea de repeticiones dispersas o en tandem, con coincidencia exacta del patrón o con presencia de "gaps"

Longitud	Cantidad	Longitud	Cantidad
1	4	11	4194304
2	16	12	16777216
3	64	13	67108864
4	256	14	268435456
5	1024	15	1073741824
6	4096	16	4294967296
7	16384	17	17179869184
8	65536	18	68719476736
9	262144	19	27487790644
10	1048576	20	1099511627776

Tabla 4.4: Cantidad de secuencias repetidas por cada longitud de un patrón.

(inserciones o deleciones) y/o mutaciones. El problema se puede plantear así:

$$descubrir(X, w, f, g, l_t)$$

donde la función "descubrir" debe encontrar una secuencia X de longitud $|X| = n$, el patrón w de longitud $|w|$ que tenga una frecuencia de aparición mayor o igual a f, para descartar secuencias con un bajo número de repeticiones; y presente un número de diferencias, incluyendo gaps, menor o igual a g. Si g es igual a cero, el patrón se buscará con una coincidencia exacta. El parámetro l_t indica la longitud de la repetición en tandem. Si l_t es igual a uno, la búsqueda se realizará para patrones repetidos de forma dispersa.

Un método para analizar repeticiones deber ser eficiente en tiempo y espacio (lineal), que sea aplicable a la mayor cantidad posible de problemas, y que provea un análisis estadístico y visualización integral de los resultados. Por último, el gran número de secuencias disponibles que incluye el genoma de varios organismos y la evidente utilidad biológica de descubrir patrones en ellas, hace importante el diseño de herramientas computacionales que ayuden en la caracterización de patrones de mayor longitud, de cuya existencia se tiene evidencia previa pero su descripción y su relevancia biológica no han sido exploradas. Podemos obtener más información sobre el estudio de secuencias patrones en [40].

4.6. Aplicación: las islas CpG

Veremos ahora una breve aplicación o ejemplo de la teoría introducida en las secciones anteriores. Se trata de observar de modo práctico algunas de las definiciones y estimaciones que hemos visto, en particular para el análisis de patrones repetidos en secuencias de ADN.

En el genoma humano, donde ocurre el dinucleótido CG (frecuentemente escrito como CpG para distinguirlo del par de bases C-G en dos cadenas), el nucleótido C normalmente se modifica químicamente por un proceso de *metilación* (adición de un grupo metilo (-CH3) a una molécula). Existe una probabilidad bastante alta de que esta metilación de C mute en una T, con la consecuencia de que, en general, los dinucleótidos CpG son más raros en el genoma de lo que podría esperarse de las probabilidades independientes de C y G. Debido a importantes razones biológicas, el proceso de metilación se suprime en determinadas extensiones cortas del genoma, como alrededor de los promotores o regiones de inicio de muchos genes. En estas regiones, observamos más dinucleótidos CpG que en ningún otro lugar, y de hecho más nucleótidos C y G en general. Tales regiones se denominan islas CpG y, en contraste, el resto del genoma es el *océano*. Normalmente están compuestas por cientos de bases de longitud. Así, las islas CpG conforman aproximadamente un 40 % de los genes promotores en mamíferos. La "p" en CpG representa que están enlazados por un fosfato. La definición formal de una isla CpG fue dada por Gardiner-Garden y Fommer ([21]), y se expresa como una región con al menos 200 pares de bases, con un porcentaje de CG mayor del 50, y con un promedio del ratio CpG "observado/esperado" mayor de 0.6. Este ratio se calcula dividiendo la proporción de dinucleótidos CpG en la región, entre lo esperado cuando se asume independencia en una distribución multinomial. La fórmula usada es

$$O/E = \frac{CpG/N}{C/N \times G/N}$$

donde N es el número de pares de bases en la secuencia considerada.

Podemos observar una secuencia de dinucleótidos pero no sabemos a qué tipo de región pertenece cada fragmento. Por ejemplo, en esta secuencia

$$\underbrace{AACATA}\underbrace{\mathbf{CGTCCG}}\underbrace{ATACATA},$$

una cuestión relevante es: dado un fragmento de la secuencia genómica, ¿cómo podemos decidir si proviene o no de una isla CpG? Parece que el fragmento señalado en negrita, $X = \mathbf{CGTCCG}$, sí podría provenir de una isla CpG.

Otra pregunta que podríamos hacernos, en general, es: dado un largo segmento de secuencia, ¿cómo podemos encontrar islas CpG en él, si hay alguna? La respuesta a esta pregunta la daremos en el siguiente capítulo, con los modelos de Markov ocultos.

Para poder modelizar las islas CpG, debemos saber primero qué tipo de peculiaridades presentan éstas y los océanos. Sabemos que

1. Hay más Ces y Ges en las islas (y más Aes y Tes en los océanos).

2. La probabilidad de hallar una G después de un nucleótido será mayor en una isla (menos en un océano) si en la posición actual hay una C que si no la hay.

En consecuencia, y como respuesta a la primera pregunta, un modelo de Markov de orden uno puede capturar estas relaciones de dependencia. Las probabilidades de cada transición van a depender de si estamos en una isla CpG o no, por lo tanto, construiremos dos modelos de Markov, uno para cada caso.

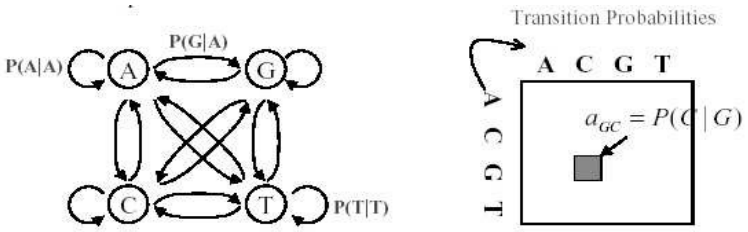

Figura 4.1: Modelo de Markov para el problema de las islas CpG.

Para poder estimar estas probabilidades, obtenemos de bancos de datos genómicos secuencias pertenecientes a islas CpG (grupo "+") y pertenecientes a océanos (grupo "−"). Las probabilidades de transición se estimarán mediante máxima verosimilitud: si n_{ij}^{*} representa el número de veces que el nucleótido j sigue al nucleótido i en una secuencia, siendo $* \in \{+,-\}$ e $i,j \in \{A,C,G,T\}$, las probabilidades de transición estimadas son

$$\widehat{p_{ij}^{+}} = \frac{n_{ij}^{+}}{\sum_{m} n_{im}^{+}}, \qquad \widehat{p_{ij}^{-}} = \frac{n_{ij}^{-}}{\sum_{m} n_{im}^{-}}, \tag{4.9}$$

de la misma forma que las obtenidas en la sección 4.1. Las siguientes tablas muestran las estimaciones de las probabilidades en dos secuencias: la primera corresponde a un modelo de isla CpG y es $S_1 =$ **CGAACAGCCTCGACATGGCGTT**; y el segundo, $S_2 =$ **AAACAGCCTGACATGGTTC**, no corresponde a una isla CpG. Evidentemente, la suma por filas en ambas tablas es uno.

Isla	A_+	C_+	G_+	T_+
A_+	0.20	0.40	0.20	0.20
C_+	0.29	0.14	**0.43**	0.14
G_+	0.33	0.33	0.17	0.17
T_+	0.00	0.33	0.33	0.33
Suma	0.82	1.21	1.13	0.84

Tabla 4.5: Estimación de las probabilidades de transición en el modelo de isla CpG.

Supongamos ahora que queremos puntuar una secuencia para decidir si corresponde a una isla CpG o a un océano, como presentábamos en la primera pregunta. Disponemos de dos modelos, como antes indicamos, el modelo "+" de las islas CpG y el modelo "−" de los océanos. La idea subyacente tras el sistema de puntuaciones es

Océano	A_	C_	G_	T_
A_	0.33	0.33	0.17	0.17
C_	0.50	0.25	**0.00**	0.25
G_	0.25	0.25	0.25	0.25
T_	0.00	0.25	0.50	0.25
Suma	1.08	1.08	0.92	0.92

Tabla 4.6: Estimación de las probabilidades de transición en el modelo de océano. Al basarse en una secuencia corta aparece un cero en la transición $C \to G$.

1. Si la secuencia pertenece a una isla CpG, tendrá una probabilidad más alta sobre el modelo "+" que sobre el "−".

2. Si la secuencia no es de una isla CpG, la probabilidad que le asignará el modelo "−" será mayor.

En lugar de multiplicar las probabilidades, sumaremos los logaritmos de las razones de probabilidades según cada modelo, y calcularemos el *logg-odds ratio* de la siguiente forma

$$S(\mathbf{X}) = log\left(\frac{P(\mathbf{X}|+)}{P(\mathbf{X}|-)}\right) = log\left(\frac{\prod_{a=1}^{n}\widehat{p}^{+}_{x_{a-1}x_a}}{\prod_{a=1}^{n}\widehat{p}^{-}_{x_{a-1}x_a}}\right) = \sum_{a=1}^{n} log\left(\frac{\widehat{p}^{+}_{x_{a-1}x_a}}{\widehat{p}^{-}_{x_{a-1}x_a}}\right) = \sum_{a=1}^{n}\beta_{x_{a-1}x_a}, \tag{4.10}$$

donde x_a denota el nucleótido en la posición a de la secuencia X, para $a = 1, \ldots, n$; y n es la longitud de la secuencia dada.

La decisión de si la secuencia es o no una isla CpG dependerá de que los valores de la suma sean más o menos altos. En la siguiente tabla se muestra un ejemplo para $i, j \in E$.

$\beta_{ij} = log(p^{+}_{ij}/p^{-}_{ij})$	A	C	G	T
A	-0.51	0.18	0.18	0.18
C	-0.56	-0.56	**1.00**	-0.56
G	0.29	0.29	-0.41	-0.41
T	1.00	0.29	-0.41	0.29

Tabla 4.7: Matriz de puntuaciones basada en los modelos de las tablas anteriores. Al tratarse en secuencias cortas, la transición $C \to G$ se puntuará con un 1. Deberían tomarse más valores para mejorar la estimación.

Ante estas puntuaciones que podríamos considerar altas (menos valores negativos que positivos), diríamos que la secuencia sí corresponde a una isla CpG. En este caso,

dada la secuencia $X = \mathbf{CGTCCG}$, se tiene una puntuación $S(X) = 1 + (-0.41) + 0.29 + (-0.56) + 1 = 1.32$. Esta puntuación normalizada por la longitud de X resulta $\frac{1.32}{6} = 0.22$. Concluimos entonces, dado que la puntuación es positiva, que sí se trata de una isla CpG, como habíamos intuido.

Capítulo 5

Los modelos de Markov ocultos

5.1. Introducción

Llegamos ya a nuestro principal objeto de estudio: los Modelos de Markov Ocultos. Utilizaremos toda la información vista en las dos partes anteriores para comprender mejor la teoría general sobre estos modelos, y sus aplicaciones prácticas.

Nuestro estudio sobre los modelos de Markov ocultos contendrá

1. Una definición de estos modelos y descripción de sus propiedades.
2. El principal algoritmo de estimación de parámetros del modelo.
3. Una aplicación y un caso práctico

Por último, introduciremos algunas de las nuevas aplicaciones de los modelos de Markov ocultos en diferentes problemas relacionados con la Biología y la Bioinformática.

Antes de comenzar con la definición, veamos una breve introducción a su historia y una idea intuitiva de dichos modelos.

Podemos decir, de manera general, que los modelos de Markov ocultos (HMMs son las siglas en inglés para "*Hidden Markov Models*") son modelos matemáticos que capturan información oculta en secuencias de símbolos u observaciones (nucleótidos en nuestro caso). Los modelos de Markov ocultos, o simplemente los modelos de Markov, nos ayudan a responder a las siguientes preguntas: "dada una secuencia de ADN, ¿pertenece a una familia particular?" o "asumiendo que la secuencia pertenece a una determinada familia, ¿qué podemos saber de su estructura interna?". Un ejemplo del segundo tipo de problema podría ser tratar de identificar determinadas regiones de una secuencia de proteína.

La mayoría de los escritos sobre los modelos de Markov ocultos pertenecen a la literatura del reconocimiento ortográfico ([31]), donde fueron aplicados por primera

vez a principios de 1970. En este campo, la aplicación de los modelos de Markov ocultos trata básicamente de, dada una señal ortográfica acústica (fonemas), reconocer la secuencia de palabras (ocultas) que han emitido tal señal, es decir, que se han dicho.

Muchos problemas en el análisis de secuencias biológicas tienen la misma estructura: basados en una secuencia de símbolos de un determinado alfabeto, encontrar qué secuencia representa. Para las proteínas, las secuencias consisten en símbolos del "alfabeto" de los 20 aminoácidos; y típicamente nosotros queremos saber a qué familia de proteínas pertenece la secuencia dada. Aquí, la secuencia primaria de aminoácidos es análoga a la señal ortográfica, y la familia de proteínas a la palabra dicha que representa. Nuestro alfabeto, como ya sabemos, consta de las iniciales de las cuatro bases de nucleótidos A, C, G y T.

En 1989, Gary Churchill ([22]) introdujo el uso de los modelos de Markov ocultos para la segmentación de secuencias de ADN. La utilización de estos modelos por Churchill, le permitió segmentar una secuencia de ADN alternando regiones de nucleótidos similares. Desde entonces, los modelos de Markov ocultos se han aplicado cada vez más a la genómica, incluyendo la búsqueda de genes y la predicción de funciones de las proteínas, sobre todo debido al trabajo pionero de David Haussler, de la universidad de California, Santa Cruz. Hoy en día, con los métodos de alineamiento, los HMMs se encuentran entre los algoritmos más representativos en el área de la Bioinformática. La necesidad básica de los HMMs surge del hecho de que los datos del genoma están inherentemente desordenados. Incluso en regiones con un alto contenido en CG, por ejemplo, puede haber largas extensiones de Aes y Ts. Los patrones observados en secuencias de ADN son necesariamente una representación "robusta" del estado oculto del genoma ("gran contenido de CG" podría ser uno de esos estados). Las cadenas de Markov simples no son lo suficientemente flexibles para capturar muchas propiedades de las secuencias de ADN. Los modelos de Markov ocultos las generalizan en un simple y eficiente esquema.

La idea básica que se encuentra detrás de la aplicación de los modelos de Markov ocultos es la de modelizar una secuencia como si hubiese sido indirectamente generada por una cadena de Markov. Cada posición en la secuencia tiene un estado desconocido (oculto) de la cadena de Markov, pero podemos observar los símbolos generados de acuerdo a una distribución multinomial que depende de ese estado. En otras palabras, la información que recibimos sobre la cadena de Markov oculta es indirecta. La secuencia que tratamos de analizar es por lo tanto modelizada siendo el resultado de un proceso doblemente estocástico: uno generando una cadena de Markov oculta, y otro transformando esta cadena oculta en una secuencia observable. Este segundo proceso sigue una distribución multinomial: en cada estado, se utiliza un conjunto diferente de parámetros para producir la secuencia observada. Una de las claves de los HMMs es tomar esta secuencia observada e inferir o deducir los estados ocultos. Éstos pueden representar diferentes tipos de secuencias. Los

modelos de Markov ocultos simples solo tienen dos estados, como "ricos en CG" o "ricos en AT", mientras que los modelos ocultos más complejos pueden tener más estados, como "A", "C", "G", "T", "región codificante" o "intrón".

Los modelos de Markov ocultos tienen dos importantes parámetros: las probabilidades de transición y las probabilidades de emisión. El parámetro de transición describe la probabilidad de que la cadena de Markov cambie a lo largo de varios estados ocultos. Estos cambios pueden ocurrir muy a menudo o de forma escasa; ya que la cadena puede hacer la transición entre solo dos estados o entre muchos estados. El parámetro de emisión describe las probabilidades de que los símbolos en la secuencia observada se produzcan en cada uno de los diferentes estados. Cada uno de los estados ocultos debe poder producir los mismos símbolos, aunque en diferentes frecuencias. El número de símbolos emitidos puede variar entre dos o muchos; el número más común en análisis de secuencias es 4 (para ADN y ARN) o 20 (para los aminoácidos).

Vamos ahora a formalizar la notación para los modelos de Markov ocultos. Necesitamos ahora distinguir la secuencia de estados de la secuencia de símbolos. Llamaremos a la secuencia de estados el *camino* π. El propio camino sigue una cadena de Markov simple, por lo tanto, la probabilidad de un estado depende solamente del estado anterior. El estado en la posición a del camino se denota por π_a. La cadena está caracterizada por los parámetros

$$p_{uv} = P(\pi_a = v | \pi_{a-1} = u),$$

que indican las *probabilidades de transición* entre estados contiguos del camino, para $a = 1, \ldots, n$, donde n es la longitud de la secuencia de observaciones que queremos analizar; y u, v son posibles estados de la cadena de Markov oculta. Para modelizar el comienzo del proceso introducimos un estado inicial I. La probabilidad de transición $p_{I,u}$ del estado inicial al estado u puede pensarse como la probabilidad de empezar en el estado u, $p_u^{(0)}$, para cualquier u. Es también posible modelizar los finales, terminando una secuencia de estados con una transición en un estado final F, $p_{u,F}$. Por convenio, tomaremos la transición al estado final como cero.

Como hemos desacoplado los símbolos i de los estados u, debemos introducir un nuevo conjunto de parámetros para el modelo, $e_u(i)$. En general, un estado puede producir un símbolo sobre todos los símbolos posibles. Por lo tanto, definimos

$$e_u(i) = P(x_a = i | \pi_a = u),$$

la probabilidad de que se observe el símbolo i cuando nos encontramos en el estado u. Éstas se conocen como las *probabilidades de emisión*. La razón por la cual las probabilidades de emisión se llaman de esta forma es porque a veces es conveniente pensar los modelos de Markov ocultos como modelos generativos, que generen o emitan secuencias. Puede generarse una secuencia de un HMM como sigue: primero se

elige un estado π_1 de acuerdo a las probabilidades $p^{(0)}_{\pi_a}$, $a = 1, \ldots, n$. En este estado, se emite una observación de acuerdo a la distribución e_{π_1} para ese estado. Después, se elige un nuevo estado π_2 en concordancia con las probabilidades de transición $p_{\pi_1\pi_a}$, $a = 1, \ldots, n$; y así sucesivamente. De esta forma, se genera una secuencia de observaciones aleatorias.

En consecuencia, cuando el HMM trabaja hay, en primer lugar, una secuencia de estados visitados, los cuales denotamos por π_1, π_2, $\pi_3, \ldots$; y en segundo lugar, una secuencia de símbolos emitidos, denotados por x_1, x_2, $x_3, \ldots$. Denotamos la secuencia entera de los π_a por Π y la secuencia de los x_a por X, y escribimos la secuencia observada como $X = x_1, x_2, \ldots$; y la secuencia de estados como $\Pi = \pi_1, \pi_2, \ldots$. Supongamos que cada símbolo x_a toma un valor del conjunto de observaciones, que es finito de tamaño k (en nuestro caso $k = 4$ para A, C, G y T); y que cada estado π_a toma uno de los valores del conjunto de estados de tamaño S. Ahora, es fácil escribir la probabilidad conjunta de una secuencia observada X y una secuencia de estados Π

$$P(X, \Pi) = p^{(0)}_{\pi_1} \prod_{a=1}^{n} e_{\pi_a}(x_a) p_{\pi_a \pi_{a+1}}, \tag{5.1}$$

donde exigimos que $p_{\pi_n \pi_{n+1}} = 0$. Sin embargo, esta expresión no es útil en la práctica porque normalmente no conocemos el camino oculto (secuencia de estados oculta). Más adelante, describiremos un algoritmo para estimar el camino encontrando la secuencia de estados más probable u óptima para el modelo de Markov oculto, llamado *algoritmo de Viterbi*.

Antes de comenzar con el estudio del algoritmo en cuestión, conviene señalar que existen un gran número de variantes de los modelos de Markov ocultos que modifican y extienden el modelo básico, para satisfacer las necesidades de diferentes aplicaciones. Por ejemplo, podemos añadir *estados silenciosos* (es decir, estados que no emiten ningún símbolo) al modelo para representar la falta de ciertos símbolos que se espera que estén presentes en posiciones específicas. Podemos también hacer que los estados emitan dos símbolos en lugar de un único símbolo y, por lo tanto, el resultante HMM genere simultáneamente dos secuencias de símbolos relacionadas. Otra variante de los modelos de Markov ocultos, adecuada para la búsqueda de genes, se denomina modelo de Markov semi-oculto o generalizado. En este caso, cada estado oculto emite una secuencia de observaciones, en lugar de una única observación o símbolo. Veremos una breve descripción de este modelo con la aplicación en búsqueda de genes. Por otro lado, es también posible hacer que las probabilidades de ciertos estados dependan de parte de las emisiones previas y así podríamos describir de forma más compleja correlaciones entre símbolos.

Veamos un ejemplo clásico y muy sencillo de cómo aplicar un modelo de Markov oculto con dos estados, para introducir una idea básica e intuitiva de éste.

Ejemplo: el casino deshonesto. Imaginemos que en un casino utilizan un dado libre o justo (correcto) la mayoría de las veces, pero otras utilizan un dado cargado. Tenemos entonces un par de dados, uno que es justo o normal, y otro que está cargado (es decir, no caemos en todas las caras con igual probabilidad). De esta forma, nuestros estados serán: {"dado libre", "dado cargado" }. Un modelo de Markov decide cuál de los dos dados juega, y dependiendo del estado del modelo, se aplicarán las probabilidades de emisión para el dado cargado o para el normal. Así, generamos una secuencia de símbolos que es el resultado de estar en dos estados diferentes. Especificaremos nuestro parámetro de transición de modo que en cualquiera de los dos estados haya un 90 % posibilidades de quedarse en ese estado, y un 10 % de cambiar de estado (el casino cambia un dado libre por uno cargado), como se muestra en la figura inferior.

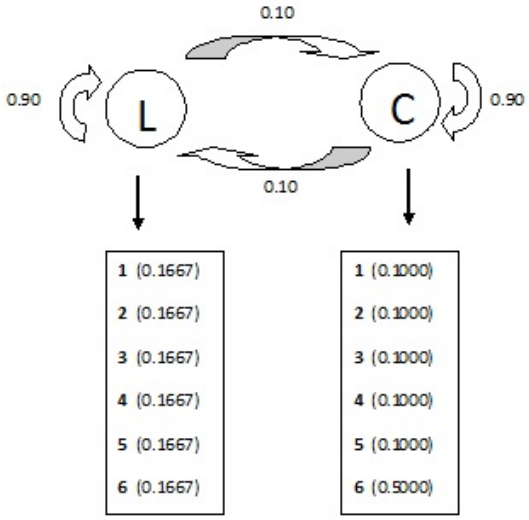

Figura 5.1: Modelo de Markov oculto asociado al ejemplo. Las transiciones entre el dado cargado y el dado libre se modelizan con una cadena de Markov, y los lanzamientos como emisiones independientes de un modelo multinomial.

Para nuestro dado no cargado o libre, la probabilidad de obtener un número entre 1 y 6 (ambos incluidos) es la misma, y está dada por

$$0.1667\, 0.1667\, 0.1667\, 0.1667\, 0.1667\, 0.1667,$$

donde cada columna describe la probabilidad para cada uno de los seis números.

Para el dado cargado, las probabilidades de emisión son

$$0.1000\, 0.1000\, 0.1000\, 0.1000\, 0.1000\, 0.5000.$$

Aquí, las probabilidades de emitir un número entre 1 y 5 son iguales, pero es mucho mayor la probabilidad de obtener un 6 (50 %).

La secuencia visible producida por tal modelo de Markov oculto podría ser como la siguiente

$$X = 45536531633635551333626651321416366516 66.$$

Si conocemos las propiedades de los dos dados y de la cadena de Markov subyacente u oculta, ¿podemos encontrar la secuencia más probable de estados ocultos detrás de X? En otras palabras, ¿podemos averiguar qué dado ha sido utilizado por el casino en cada instante o posición de la secuencia de resultados? Más adelante, con el algoritmo de Viterbi, veremos cómo responder a estas preguntas, aquí simplemente mostramos la secuencia oculta que genera nuestra secuencia visible

$$Oculto: \ \Pi = 11111111111111111111222211111122222222 2.$$

$$Visible: \ X = 45536531633635551333626651321416366516 66.$$

Observemos que el símbolo "6" ocurre con una frecuencia mayor cuando el estado en la secuencia oculta es 2, correspondiente al dado cargado, pero esta dependencia es simplemente probabilística. En las aplicaciones biológicas, aplicaremos nuestro modelo de Markov oculto en un conjunto de datos donde los estados ocultos son conocidos, incluso cuando no sepamos exactamente cuales son las probabilidades de transición y emisión. Esto permite calcular las matrices de transición y emisión de nuestro modelo basadas en los datos y, por lo tanto, una mejor inferencia en los estados ocultos de nuevos datos. Este aspecto especialmente útil en la búsqueda de genes, donde los modelos se aplican a genomas conocidos para después extenderlos a otros genomas semejantes.

Seguimos ahora con el ejemplo de las islas CpG dado en el capítulo anterior (sección 4.6), para estudiar su análisis con los modelos de Markov ocultos.

Aplicación: las islas CpG. Recordemos la segunda pregunta planteada relativa al análisis de las islas CpG: ¿cómo podemos encontrarlas, si las hay, en una larga secuencia no conocida? Los modelos de cadenas de Markov ya descritos pueden utilizarse para responder esta cuestión, calculando las puntuaciones log-odds de un conjunto o "ventana" de, por ejemplo, 100 nucleótidos "alrededor" de cada nucleótido en la secuencia. Así, las puntuaciones altas indicarían islas CpG potenciales. Sin embargo, esto no nos vale ya que, de hecho, las islas CpG tienen unos límites definidos y son de longitud variable. Por lo tanto, el problema sería saber qué tamaño de ventana hay que usar. Una aproximación más satisfactoria sería construir un modelo simple de la secuencia entera, no sólo 100 nucleótidos, que incorpore ambas cadenas de Markov, el modelo "+" de las islas CpG, y el modelo "−" del resto de regiones (océano). Para ello, utilizaremos un modelo de Markov oculto.

Para simular en un mismo modelo las "islas" en un "océano", queremos tener ambos modelos de Markov presentes en uno solo, con una pequeña probabilidad de cambiar de un modelo al otro en cada punto de transición. Sin embargo, esto

introduce las posibles dificultades de tener dos estados con el mismo nombre que correspondan a cada nucleótido. Resolveremos este problema etiquetando de nuevo los estados. Ahora tenemos los estados A_+, C_+, G_+ y T_+ en las regiones de islas CpG, y los estados A_-, C_-, G_- y T_- correspondientes a las regiones de océano. Cada uno de los estados de ambos modelos puede emitir uno de los cuatro símbolos de nuestro alfabeto A, C, G y T. De esta forma, además de las probabilidades de transición entre los dos grupos (isla y océano), hay también un conjunto de transiciones en cada uno de los grupos, como en los modelos de Markov simples. Las probabilidades de transición en este modelo están establecidas de modo que los elementos dentro de cada grupo tienen probabilidades de transición mayores. En general, es más probable cambiar de "+" a "−" que al revés (hay más regiones de océano que de isla en el genoma), por tanto, si lo recorremos libremente, el modelo pasará más tiempo en los estados "−" de océano que en los estados de islas. El etiquetado es el paso importante. Las probabilidades de emisión en este caso son todas 0 o 1, dependiendo de si el símbolo emitido coincide con el estado (tanto positivo como negativo) o no. Por ejemplo, $P(x_a = C|\pi_a = C_-) = 1$ y $P(x_a = C|\pi_a = A_+) = 0$.

Así, siguiendo la ecuación 5.1, podemos calcular la probabilidad de la secuencia **CGCG** que ha sido emitida por la secuencia de estados (C_+, G_-, C_-, G_+)

$$p^{(0)}_{C_+} \times 1 \times p_{C_+,G_-} \times 1 \times p_{G_-,C_-} \times 1 \times p_{C_-,G_+} \times 1 \times 0.$$

La diferencia esencial entre una cadena de Markov y un modelo de Markov oculto es que para este último no hay una correspondencia entre los estados y los símbolos. Es decir, no es posible, a la larga, saber en qué estado está el modelo cuando se genera el símbolo x_a, sólo observando x_a. En nuestro ejemplo, no hay forma posible de saber, observando sólo un símbolo C, si fue emitido por un estado C_+ o C_-. Por ello, la fórmula 5.1 no es válida para nuestro interés, y tendremos que estudiar un algoritmo que estime la secuencia de estados ocultos.

5.2. El algoritmo de Viterbi

A pesar de que no es posible saber en qué estado está el sistema observando los símbolos correspondientes, sí podemos estimar la secuencia de estados oculta. Describimos a continuación el algoritmo de programación dinámica más común para poder hallarla, denominado *algoritmo de Viterbi*.

En general, es posible que haya muchas secuencias de estados que pueden dar lugar a una particular secuencia de símbolos. Por ejemplo, en nuestro modelo de islas CpG, las secuencias de estados (C_+, G_+, C_+, G_+), (C_-, G_-, C_-, G_-) y (C_+, G_-, C_+, G_-) podrían generar la secuencia de símbolos **CGCG**. Sin embargo, lo hacen con diferentes probabilidades. La tercera es el producto de múltiples probabilidades alternando entre los dos modelos (del modelo "+" al "−", del "−" al "+", y así sucesivamente), y por lo tanto es mucho más pequeña que la probabilidad de las dos

primeras (las probabilidades de transición entre modelos son menores). La segunda es significativamente más pequeña que la primera, ya que contiene dos transiciones de C a G que son significativamente menos probables en el caso del modelo "−" de océanos, que en el modelo "+" de islas. Aplicaremos el algoritmo de Viterbi para hallar la sucesión de estados más probable de haberla generado. La estimación de un camino a través de un modelo de Markov oculto nos dirá qué parte de la secuencia se predice por una isla CpG, ya que anteriormente asumimos que cada estado se asignaba al modelo de las islas CpG o al del océano. Si tenemos que elegir sólo un camino, el que tenga la mayor probabilidad deberá ser el escogido.

De entre todas las posibles secuencias de estados Π, queremos encontrar la secuencia de estados que explique mejor la secuencia de símbolos observada. Dicho de otra forma, queremos encontrar "el mejor alineamiento" entre la secuencia de símbolos y el HMM, por lo que a veces este procedimiento se denomina *problema del alineamiento óptimo* o *decodificación*. Formalmente, dada una secuencia observada $X = x_1, x_2, x_3, ..., x_R$ de salidas u observaciones, queremos computar eficientemente una secuencia de estados $\Pi = \pi_1, \pi_2, \pi_3, ..., \pi_R$ que tenga la mayor probabilidad condicionada dada X (tomamos ahora R en lugar de n como longitud de la secuencia para facilitar la notación). En otras palabras, queremos encontrar el camino óptimo Π^* que satisfaga lo siguiente

$$\Pi^* = argmax_{\Pi} P(\Pi|X). \tag{5.2}$$

Encontrar la secuencia de estados óptima Π^* comparando todas las S^R (S el el tamaño del conjunto formado por todos los estados) posibles secuencias de estados es computacionalmente inviable. Sin embargo, podemos utilizar un algoritmo de programación dinámica, llamado *algoritmo de Viterbi*, para encontrar una de las muchas secuencias de estados que pueden maximizar la probabilidad anterior. El algoritmo se divide en dos partes: primero encuentra el $max_{\Pi} P(\Pi|X)$, y después realiza un procedimiento "hacia atrás" para encontrar la secuencia Π que satisfaga este máximo.

En primer lugar se define, para r y u arbitrarios,

$$\delta_r(u) = max_{\pi_1, \pi_2, ..., \pi_{r-1}} P(\pi_1, \pi_2, ..., \pi_{r-1}, \pi_r = u; x_1, x_2, ..., x_r), \tag{5.3}$$

($\delta_1(u) = P(\pi_1 = u; x_1)$). En otras palabras, $\delta_r(u)$ es la máxima probabilidad de todas las formas de terminar en el estado u en el instante o posición r; y haber observado la secuencia $x_1, x_2, ..., x_r$. Entonces

$$max_{\Pi} P(\Pi, X) = max_u \delta_R(u). \tag{5.4}$$

La probabilidad en esta expresión es la probabilidad conjunta de Π y X, no una probabilidad condicionada. Nuestro objetivo es encontrar la secuencia Π para la cual se alcance la máxima probabilidad condicionada. Ya que

$$max_{\Pi} P(\Pi|X) = max_{\Pi} \frac{P(\Pi, X)}{P(X)}, \tag{5.5}$$

y ya que el denominador del lado derecho de la igualdad no depende de Π, se tiene que

$$argmax_{\Pi} P(\Pi|X) = argmax_{\Pi} \frac{P(\Pi, X)}{P(X)} = argmax_{\Pi} P(\Pi, X). \tag{5.6}$$

El primer paso es calcular los $\delta_r(u)$ inductivamente en r. Después, con el procedimiento "hacia atrás", recuperaremos la secuencia que de el mayor $\delta_R(u)$. El paso de inicialización es

$$\delta_1(u) = p_u^{(0)} e_u(x_1), \qquad 1 \leq u \leq S. \tag{5.7}$$

teniendo en cuenta que $\delta_0(0) = 1$ y $\delta_r(0) = 0$, para todo $r > 0$.

El paso de inducción es

$$\delta_r(v) = max_{1 \leq u \leq S}\{\delta_{r-1}(u) p_{uv}\} e_v(x_r), \qquad 2 \leq r \leq R,\ 1 \leq v \leq S. \tag{5.8}$$

Para obtener la observación con la máxima probabilidad, recuperamos los π_a como sigue. Se define

$$\psi_R = argmax_{1 \leq u \leq S} \delta_R(u). \tag{5.9}$$

Y sea $\pi_R = \psi_R$. Entonces, π_R es el estado final en la secuencia de estados requerida. Los restantes π_r para $r \leq R - 1$ se encuentran recursivamente definiendo primero

$$\psi_r = argmax_{1 \leq u \leq S}\{\delta_r(u) p_{u\psi_{r+1}}\}. \tag{5.10}$$

Y después escribiendo $\pi_r = \psi_r$. Si el máximo no es único, tomaremos arbitrariamente un valor de u.

El algoritmo de Viterbi encuentra la secuencia de estados óptima con una eficiencia del orden de $O(RS^2)$.

Veremos cómo obtener el camino oculto en un problema de predicción de islas CpG utilizando el algoritmo de Viterbi en el siguiente capítulo, donde analizaremos un ejemplo con R.

Existen otros dos algoritmos que resuelven problemas relacionados con un HMM, como el *problema de puntuación* o *evaluación* y el *problema de estimación*. El primero de ellos, consiste en calcular la probabilidad $P(X)$, dada la secuencia observada $X = x_1, \ldots, x_R$, basada en un modelo de Markov oculto. Un método eficiente para calcular dicha probabilidad está basado en un algoritmo de programación dinámica, llamado *algoritmo "hacia delante"* (existe un análogo llamado *algoritmo "hacia atrás"*). Este algoritmo tiene un tiempo de computación del orden de $O(RS^2)$. Si queremos clasificar una nueva proteína, podemos construir un HMM para cada familia conocida de proteínas y utilizarlo para calcular la probabilidad de la nueva secuencia. Asignaremos la proteína a la familia en cuyo modelo se obtenga mayor probabilidad. En cuanto al segundo problema, se refiere a cómo podemos elegir los

parámetros de un HMM basándose en un conjunto de secuencias observadas. No existe una manera óptima de estimar estos parámetros de un número limitado de secuencias de observaciones finitas, pero sí existen formas de localizar los parámetros del modelo que maximicen localmente la probabilidad de la secuencia observada. Por ejemplo, el *algoritmo de Baum-Welch* estima y actualiza de forma iterativa el conjunto de parámetros del modelo basado en un procedimiento *"hacia delante-hacia atrás"*. Existen muchos métodos para la estimación de parámetros de un HMM, ya que en el fondo se trata de un problema de optimización. No profundizaremos en el estudio de ambos problemas y algoritmos ([40], [10]).

Ejemplo: el casino deshonesto Veamos cómo aplicar el algoritmo de Viterbi a nuestro ejemplo del casino deshonesto. Recordemos que la probabilidad de quedarse en cualquiera de los dos estados (dado libre o cargado) es 0.9, y la probabilidad de cambiar de estado es 0.1. Además, la probabilidad de obtener un número en el dado libre es siempre la misma ($\frac{1}{6}$), mientras que en el dado cargado la probabilidad de que salga un 6 es 5 veces mayor que la del resto.

Con estos datos, y teniendo en cuenta el diagrama presentado en la figura 5.1, veamos cómo aplicar el algoritmo de Viterbi a una secuencia de lanzamientos, como por ejemplo $X = 6\,2\,6$, correspondiente a 3 lanzamientos.

El paso inicial es $1\,0\,0$. Después, para el símbolo 6 se tiene

$$\frac{1}{2} \times \frac{1}{6} = \frac{1}{12}$$
$$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4} = \mathbf{0.25}$$

siguiendo con el siguiente número obtenido, el 2, se obtiene

$$\frac{1}{6} \times max\{(\frac{1}{12} \times 0.9, \frac{1}{4} \times 0.1) = 0.0125$$
$$\frac{1}{10} \times max\{(\frac{1}{12} \times 0.1, \frac{1}{4} \times 0.9) = \mathbf{0.0225}$$

y finalmente aplicamos el algoritmo para el último símbolo, de nuevo el 6.

$$\frac{1}{6} \times max\{(0.0125 \times 0.9, 0.0225 \times 0.1) = 0.000375$$
$$\frac{1}{2} \times max\{(0.0125 \times 0.1, 0.0225 \times 0.9) = \mathbf{0.005625}$$

Las primeras filas representan los cálculos para el estado "dado libre" y las segundas para el estado "dado cargado". Así, tomando los mayores valores de δ, que se presentan en negrita, obtenemos que el camino óptimo es $\Pi = \{222\}$ para el conjunto de símbolos $X = 6\,2\,6$ (el estado 2 hace referencia al dado cargado). Se procedería de forma análoga para un conjunto de símbolos mayor, es decir, para más lanzamientos de los dados.

Ejemplo: reconocimiento de regiones en secuencias de ADN. Otro ejemplo sencillo con tres estados y aplicado al estudio del ADN es el siguiente. Imaginemos

que tenemos un problema de reconocimiento de una región o posición de empalme 5' ("sitio donor 5"), en una secuencia genética. Asumiremos que nos dan la secuencia de ADN que comienza con un exón, contiene un lugar de empalme 5' y termina con un intrón. El problema es identificar donde se produce el cambio del exón al intrón, es decir, donde está el lugar de empalme 5'. Sabemos que las secuencias de exones, los lugares de empalme y los intrones tienen diferentes propiedades estadísticas. Imaginemos algunas diferencias simples: supongamos que los exones tienen, en promedio, una composición de base uniforme (25 % cada base), los intrones son ricos en *A*es y *T*s (es decir, 40 % cada una para *A* y *T*, 10 % cada una para *C* y *G*), y los lugares de empalme 5' tienen como nucleótido mayoritario a *G* (95 % de *G*s y 5 % de *A*es). Con esta información, podemos dibujar el modelo de Markov oculto asociado, como se muestra en la figura inferior.

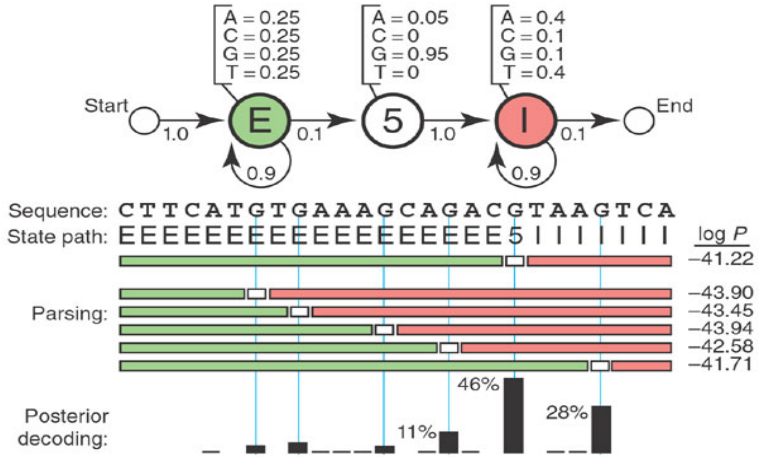

Figura 5.2: Ejemplo de un modelo de Markov oculto para el reconocimiento de una región de empalme 5'.

El modelo tiene tres estados (más un estado inicial y otro final), uno para cada uno de los tres niveles que podemos asignar a un nucleótido: E (exón), 5 (empalme 5') e I (intrón). Así, el conjunto de estados es $\Pi = \{E, 5, I\}$ y el conjunto de observaciones son los cuatro nucleótidos $X = \{A, C, G, T\}$. Cada estado tiene sus propias probabilidades de emisión (mostradas debajo de los estados), que modelan la composición básica de los exones, intrones y las *G*s en los lugares de empalme. Cada estado tiene también sus probabilidades de transición (flechas): las probabilidades de moverse de ese estado a un nuevo estado, desde el estado inicial y hacia el estado final (en este caso las probabilidades de transición de los estados inicial y final son no nulas). Las probabilidades de transición describen el orden lineal en el cual esperamos que los estados ocurran: uno o más Es, un 5, uno o más Ies.

Consideremos la secuencia mostrada en la figura, compuesta por 26 nucleótidos, y el camino de estados correspondiente, donde hay 27 transiciones y 26 emisiones en total. Multiplicando todas las 53 probabilidades juntas (y tomando el logaritmo, ya que se trata de números pequeños), podemos calcular la $P(X, \Pi) = -41.22$.

Existen otros 14 posibles caminos que no tienen probabilidad cero, ya que el lugar de empalme 5' debe estar en una de las 14 *A*es o *G*s internas. En la figura se enumeran los seis caminos con las mayores puntuaciones (con G en el lugar de empalme). El mejor tiene el valor de -41.22 para el logaritmo de la probabilidad, lo que indica que la posición más probable para el empalme es el quinto G. El camino alternativo difiere solo ligeramente de éste, -41.71 frente a -41.22. Teniendo en cuenta esta leve diferencia, ¿cómo de seguros estamos de que la quinta G es la elección correcta? Aquí se muestra la ventaja del modelado probabilístico: podemos calcular nuestra fiabilidad directamente. La probabilidad de que un estado emita un aminoácido i es la suma de las probabilidades de todos los caminos que utilizan ese estado para generar tal aminoácido, normalizada por la suma de todos los posibles caminos. En este ejemplo, solo hay un camino de estados en el numerador, y una suma de 14 caminos en el denominador. Obtenemos una probabilidad del 46 % de que la mejor puntuación de la quinta G es correcta, y un 28 % de que la sexta G es correcta (parte inferior de la figura). A ésto se le denomina *decodificación posterior*. Para problemas grandes, se utilizan algoritmos de programación dinámica, como el algoritmo *forward* o *"hacia deltante"*.

5.3. Aplicaciones: búsqueda de genes

Hoy en día, se producen ya secuencias de genes con longitudes del orden de millones de bases. Tales secuencias consisten en la colección de genes separados unos de otros por largas extensiones de secuencias no funcionales. Es de vital importancia encontrar donde están los genes en la secuencia y, por lo tanto, son muy útiles métodos computacionales que identifiquen rápidamente una gran proporción de los genes. El problema implica tomar al mismo tiempo una gran cantidad de diversa información, y se han estudiado muchas aproximaciones para poder hacerlo. Actualmente, un exitoso y popular buscador de genes para secuencias de ADN humano es el GENSCAN ([9]), que está basado en una generalización de los modelos de Markov ocultos. Expondremos a continuación, de manera breve, un algoritmo similar con el espíritu del GENSCAN para ilustrar los conceptos básicos de un HMM en la búsqueda de genes humanos. Para incrementar la precisión del procedimiento es necesario introducir algunos detalles que no describiremos aquí ([40]).

Además de la herramienta GENSCAN, existen muchos más sistemas para la predicción de genes, tanto en organismos eucariotas como procariotas. En la siguiente dirección web se describen algunos de los más utilizados y precisos:
http://cmgm.stanford.edu/classes/genefind/.

Antes de comenzar con la introducción al algoritmo en el que se basa la herramienta GENSCAN, veamos un ejemplo intuitivo que nos muestra la idea y objetivo del posterior estudio.

Ejemplo: modelos de Markov ocultos para genes eucariotas. Los modelos de Markov ocultos pueden utilizarse efectivamente para representar secuencias biológicas, como ya hemos visto. Como un ejemplo simple, consideremos un modelo de Markov oculto que modele genes codificantes en proteínas de organismos eucariotas. Es conocido que muchas regiones codificantes a proteínas muestran tendencias entre codones. El uso no uniforme de los codones resulta en diferentes símbolos estadísticos para diferentes posiciones de los codones. Estas propiedades no se observan en los intrones, ya que no se traducen en aminoácidos. Por lo tanto, es importante incorporar estas características de los codones a la hora de modelar genes codificantes en proteínas y construir un buscador de genes. La figura inferior muestra un ejemplo de un HMM para modelizar genes eucariotas. El modelo de Markov oculto dado, trata de capturar las diferencias estadísticas en exones e intrones. El modelo tiene cuatro estados, donde E_1, E_2, y E_3 se utilizan para modelar las propiedades estadísticas básicas en los exones. Cada estado E_a utiliza un conjunto de probabilidades de emisión diferentes para reflejar el símbolo en la posición a de un codón. El estado I se utiliza para modelizar los intrones. Observemos que este HMM puede representar genes con múltiples exones, donde los respectivos exones pueden tener un número variable de codones; y los intrones pueden tener también longitudes variables. Este ejemplo muestra que si conocemos la estructura y las características importantes de la secuencia biológica de interés, construir el modelo de Markov oculto es relativamente sencillo y puede hacerse de una manera intuitiva.

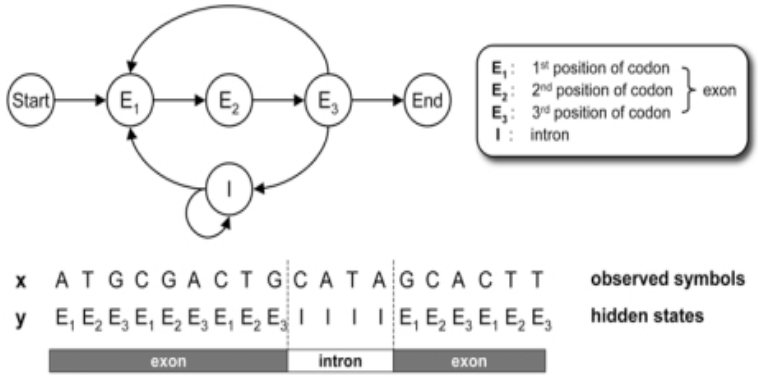

Figura 5.3: Un modelo de Markov oculto sencillo para modelizar genes eucariotas.

El HMM construido se utiliza ahora para analizar nuevas secuencias observadas. Por ejemplo, supongamos que tenemos una nueva secuencia de ADN $X = x_1, \ldots, x_{19} =$ **ATGCGACTGCATAGCACTT**. ¿Cómo podemos saber si esta secuencia de ADN es la región codificada de un gen o no? O, si asumimos que X es un gen que codifica en proteína, ¿cómo podemos predecir las posiciones de los exones y los intrones en la secuencia dada? Podemos responder a la primera pregunta calculando la probabilidad observada de X basada en el HMM dado, que modeliza genes codificantes. Si esta probabilidad es alta, implica que esta secuencia de ADN es probablemente la de la región codificada de un gen. Por otra parte, podríamos

concluir que X no es probable que sea una región codificante ya que no contiene las propiedades estadísticas que se observan típicamente en los genes codificantes en proteínas. La segunda cuestión trata de la predicción de la estructura interna de la secuencia, que no puede ser directamente observada. Para responder a esta pregunta, debemos primero predecir la secuencia de estados Π en el modelo de Markov oculto que mejor describa X. Una vez que tenemos el mejor Π, es sencillo predecir las localizaciones de los exones e intrones. Por ejemplo, supongamos que la secuencia de estados óptima es Π, como se muestra en la figura (donde se denomina y). Esto implica que las nueve primeras bases $x_1, \ldots, x_9$ pertenecen al primer exón, las siguientes cuatro bases $x_{10}, \ldots, x_{13}$ pertenecen a un intrón, y las últimas seis bases $x_{14}, \ldots, x_{19}$ pertenecen al otro exón.

5.3.1. GENSCAN

El sofware para la búsqueda de genes GENSCAN (http://genes.mit.edu/GENSCAN.html) ha sido desarrollado por Chris Burge y Samuel Karlin en la Universidad de Stanford, y está actualmente alojado en el departamento de biología del MIT (Instituto Tecnológico de Massachusetts). Este programa utiliza un modelo probabilístico complejo sobre la estructura del gen, que se basa en la actual información biológica sobre las propiedades de las señales de transcripción, traducción y de empalme. Además, utiliza varias propiedades estadísticas de regiones codificantes y no codificantes. Para tener en cuenta la heterogeneidad del genoma humano que afecta a la estructura y densidad de los genes, GENSCAN deriva diferentes conjuntos de modelos de genes de regiones del genoma con diferente contenido en CG. Es muy rápido y su precisión hace que el GENSCAN sea el método elegido para el análisis inicial de grandes (en el rango de megabases) conjuntos de ADN en genomas eucariotas. GENSCAN ha sido utilizado como la herramienta principal para la predicción de genes en el Proyecto Internacional del Genoma Humano.

A continuación, exponemos la idea para poder responder a la segunda cuestión del ejemplo, y por tanto poder predecir estructuras de genes. Como hemos dicho, el siguiente algoritmo está basado en el espíritu del GENSCAN.

Modelos de Markov semi-ocultos

Un *modelo de Markov semi-oculto* (semiHMM) o *generalizado* es un modelo de Markov oculto en el que cada estado puede emitir una secuencia de observaciones, y no solamente una observación o símbolo como en los modelos de Markov ocultos estudiados. Veamos cómo podemos modelizar esta idea.

En un HMM, supongamos que p es la probabilidad de transición de cualquier estado a él mismo. La probabilidad de que el proceso se quede en un estado para n pasos es $p^{(n-1)}(1-p)$ y, por lo tanto, la cantidad de tiempo que el proceso está en este estado sigue una distribución geométrica de parámetro $(1-p)$. Para el modelo de ge-

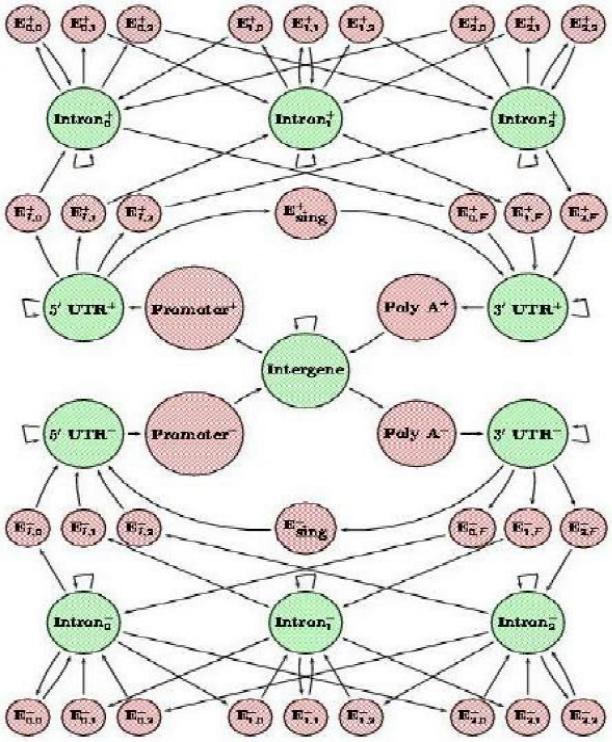

Figura 5.4: Diagrama de estados del HMM en el que se basa GENSCAN.

nes que construyamos, es necesario permitir otras distribuciones para esta cantidad. En un HMM semi-oculto (semiHMM) todas las probabilidades de transición de un estado a él mismo son cero; y cuando el proceso visita un estado, éste produce no un sólo símbolo del alfabeto sino una secuencia entera. La longitud de la secuencia puede seguir cualquier distribución, y el modelo que genera la secuencia de esa longitud puede seguir también cualquier distribución. Las posiciones en la secuencia emitida de un estado no necesitan ser iid. El modelo se formula más precisamente como sigue.

Cada estado π tiene asociado a él mismo una variable aleatoria L_π (L de "longitud") cuyo rango es un subconjunto de $0, 1, 2, \ldots$; y para cada valor observado l de L_π existe una variable aleatoria $Y_{\pi,l}$ cuyo rango consiste de todas las secuencias de longitud l. Cuando se visita el estado π, se determina aleatoriamente una longitud l de la distribución de L_π. Luego, la distribución de $Y_{\pi,l}$ se utiliza para determinar una secuencia de longitud l. Después, se toma una transición a un nuevo estado y el proceso se repite, generando otra secuencia. Estas secuencias se concatenan para crear la secuencia final del modelo de Markov semi-oculto.

Los algoritmos involucrados en este modelo tienen un orden de magnitud más complejo que un HMM regular, ya que dada una secuencia observada no sólo no conocemos el camino de estados que produce, sino tampoco conocemos la división de puntos en la secuencia indicando dónde se hace una transición a un nuevo estado.

La aplicación en la búsqueda de genes requiere una generalización del algoritmo de Viterbi. Es una generalización natural, sin embargo, ya que se trabaja generalmente con secuencias muy largas, la generalización natural no funciona en tiempo razonable. En la práctica, deben hacerse más suposiciones. Burge ([9]) observó que si las longitudes de las largas regiones intergénicas se toman siguiendo distribuciones geométricas, y si estas longitudes generan secuencias en una manera relativamente iid, entonces el algoritmo puede ajustarse de modo que puedan obtenerse tiempos de funcionamiento prácticos. Estas suposiciones no son irrazonables en nuestro caso y, por tanto, no deberían afectar en gran medida a la precisión de las predicciones. Omitiremos los detalles técnicos que subyacen en este tema. Nuestro objetivo es mostrar la idea principal de cómo trabaja un modelo de Markov oculto buscador de genes.

De esta forma, un par ϕ es una secuencia de estados $\pi_1, \pi_2, ..., \pi_n$ y una secuencia de longitudes $l_1, l_2, ..., l_n$. Dada una secuencia observada X de un semiHMM, el algoritmo de Viterbi encuentra un par óptimo ϕ_{opt} tal que la $P(\phi_{opt}|X) \geq P(\phi|X)$ para todos los pares ϕ. En otras palabras, ϕ_{opt} es el par más probable para dar lugar a la secuencia X. El par óptimo da las predicciones de genes, así como la predicciones de la estructura completa del gen.

En la práctica, hay muchas más consideraciones para optimizar el método. Quizás una de las más importantes es que muchas de las probabilidades estimadas dependen del contenido CG de la región de ADN buscada. Por ejemplo, regiones con mayor contenido de CG (ilas CpG) tienden a contener una densidad de genes significativamente mayor. El modelo de Burge toma esto y otros factores en cuidadosa consideración, y continúa refinándose tanto como diferentes tipos de datos haya disponibles.

Veamos de forma breve la idea de modelización de los modelos de Markov semiocultos.

El modelo. Cada estado del modelo que construyamos es un modelo es su propio sentido. Es necesario "entrenar" a cada estado para producir secuencias que modelen las partes correspondientes del gen en cuestión. Para ello, utilizamos un gran conjunto de datos de entrenamiento que consiste en largas cadenas de ADN, donde las estructuras de los genes han sido completamente caracterizadas. Burge et al. ([9]) recopilaron 2.5 millones de pares de bases (Mb) de ADN humano con 380 genes, 142 genes exón individuales, y un total de 1492 exones y 1254 intrones. Además, se incluyeron regiones sólo codificantes (sin intrones) de 1619 genes humanos.

Modelamos, por ejemplo, un gen orientado de 5' a 3', con 13 estados de un modelo de Markov semi-oculto, como muestra la figura inferior. La primera fila representa la región intergénica o no funcional. La segunda fila representa la región promotora. La tercera fila es la 5'UTR. La cuarta fila de los cinco estados representa los intrones y exones. La fila final es la 3'UTR y la señal Poli(A). La región no funcional entre

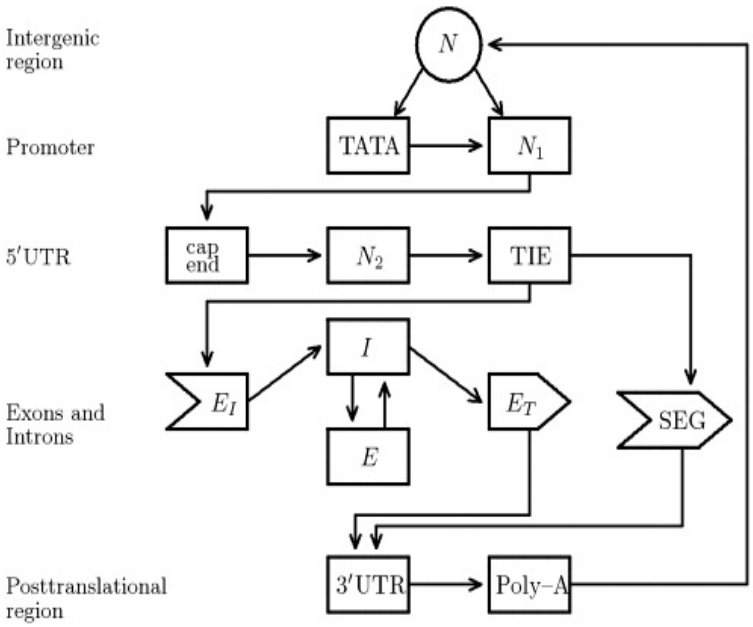

Figura 5.5: Diagrama de estados.

genes está etiquetada por N en la figura.

Primero se describe cómo se forma cada estado dando las distribuciones de L_π y de $Y_{\pi,l}$ para cada estado π. Después, se establecen las probabilidades de transición entre estados basadas de los modelos elegidos para cada región del gen. Para cada uno de los estados tendremos un modelo distinto, dependiendo de sus características genéticas. La figura muestra la complejidad de esta modelización, y las consecuentes estimaciones de parámetros de forma óptima. No profundizaremos más en el estudio de los modelos de Markov semi-ocultos en búsqueda de genes por no ser objeto de estudio del presente trabajo ([40]).

5.4. Caso práctico: las islas CpG

Como aplicación práctica a toda la teoría estudiada anteriormente, y siguiendo con los ejemplos ya propuestos sobre este tema, nos planteamos la búsqueda de islas CpG en un organismo en particular, el "Danio Rerio" o "pez cebra". Tomaremos la secuencia de datos del cromosoma 10 de tal organismo, para analizarla y encontrar en ella las posibles islas CpG, utilizando los modelos de Markov ocultos. Realizaremos esta búsqueda con R y el programa *Bioconductor*, de donde podemos descargarnos el genoma entero del pez cebra. Finalmente, una vez encontradas las posibles islas, comprobaremos si éstas se encuentran en las mismas posiciones, aproximadamente, que las estudiadas en el artículo "*Redefining CpG islands using hidden Markov models*", por Rafael A. Irizarry et al. ([24]); donde se utiliza una variante "más fina"

de los modelos de Markov ocultos, llamados *modelos de Markov ocultos jerárquicos*. Estas posiciones ya estudiadas serán nuestros datos de entrenamiento. Podemos descargárnoslos de la siguiente dirección web: http://rafalab.jhsph.edu/CGI/. El paquete de R "makeCGI" realiza la búsqueda de regiones CpG automáticamente, pero no lo utilizaremos para el presente trabajo pues nos interesa programar los pasos y el algoritmo utilizado.

Comenzamos por descargarnos el programa Bioconductor y el genoma del pez cebra (versión 6 correspondiente a la secuenciación de su genoma en el año 2008), incluyendo en R las sentencias

```
> source("http://bioconductor.org/biocLite.R")
> available.genomes() # genomas disponibles de diferentes organismos
[16] "BSgenome.Drerio.UCSC.danRer5"
[17] "BSgenome.Drerio.UCSC.danRer6"
[18] "BSgenome.Drerio.UCSC.danRer7"

> biocLite("BSgenome.Drerio.UCSC.danRer6") # versión 6, como el artículo
> library(BSgenome.Drerio.UCSC.danRer6)
> Drerio
Zebrafish genome
|
| organism: Danio rerio (Zebrafish)
| provider: UCSC
| provider version: danRer6
| release date: Dec. 2008
| release name: Sanger Institute Zv8
|
| single sequences (see '?seqnames'):
|   chr1   chr2   chr3   chr4   chr5   chr6   chr7   chr8
|   chr9   chr10  chr11  chr12  chr13  chr14  chr15  chr16
|   chr17  chr18  chr19  chr20  chr21  chr22  chr23  chr24
|   chr25  chrM
|
| multiple sequences (see '?mseqnames'):
|   Zv8_NA        Zv8_scaffold  upstream1000  upstream2000
|   upstream5000
|
| (use the '$' or '[[' operator to access a given sequence)

> class(Drerio) # objeto clase BSgenome
[1] "BSgenome"
attr(,"package")
[1] "BSgenome"
```

```
> seqnames(Drerio) # elegiremos el cromosoma 10 etiquetado por ‘‘chr10’’
 [1] "chr1"  "chr2"  "chr3"  "chr4"  "chr5"  "chr6"
 [7] "chr7"  "chr8"  "chr9"  "chr10" "chr11" "chr12"
[13] "chr13" "chr14" "chr15" "chr16" "chr17" "chr18"
[19] "chr19" "chr20" "chr21" "chr22" "chr23" "chr24"
[25] "chr25" "chrM"

> crom10<-Drerio$chr10
> crom10
  43467561-letter "MaskedDNAString" instance (# for masking)
seq: CACACACACACACACACACACACA...TATTCGGAACATTTTAATGTCAGAT
masks:
  maskedwidth  maskedratio active names
1      110500 2.542126e-03   TRUE AGAPS
2         413 9.501338e-06   TRUE   AMB
3    22431376 5.160486e-01  FALSE    RM
4     1151720 2.649608e-02  FALSE   TRF
                               desc
1                     assembly gaps
2           intra-contig ambiguities
3                       RepeatMasker
4 Tandem Repeats Finder [period<=12]
all masks together:
  maskedwidth maskedratio
     22573123   0.5193096
all active masks together:
  maskedwidth maskedratio
       110913 0.002551627

> length(crom10) # 43.467.561 de nucleótidos de longitud
[1] 43467561
```

Una vez tenemos el cromosoma 10 disponible, observamos lo que serán nuestros datos de entrenamiento, es decir, las posiciones ya estudiadas de las islas que nos hemos descargado.

```
> posic<-read.table(file="danRer6.txt",header=TRUE)
> head(posic)

  ï..chr start   end length CpGcount GCcontent pctGC obsExp
1  chr10 10146 10266    121       11        72 0.595  1.034
2  chr10 11319 12498   1180      148       731 0.619  1.312
3  chr10 25650 25762    113       10        56 0.496  1.471
4  chr10 34657 34818    162        8        84 0.519  0.756
5  chr10 45703 45986    284       28       157 0.553  1.290
```

```
6  chr10 46944 47514    571       38       301 0.527  0.967
```

```
> posic10<-subset(posic,posic[,1]=="chr10") # solo el cromosoma 10
> dim(posic10) # es un data.frame de 3889 filas 8 columnas
[1] 3889    8

> start<-posic10[,2] # comienzo isla
> end<-posic10[,3] # final isla
```

Las columnas del objeto "posic10" indican, respectivamente: el cromosoma en cuestión, las posiciones iniciales de las islas, las posiciones finales, la longitud de las regiones de islas, el ratio de islas (número de CpGs relativo a la cantidad de oportunidades de islas CpGs), el número de dinucleótidos CG en cada isla, el porcentaje de dinucleótidos CG, y el promedio del ratio "observado/esperado".

Si nos fijamos en la clase de nuestro objeto "crom10", obtenemos lo siguiente

```
> class(crom10) # objeto de la clase MaskedDNAString
[1] "MaskedDNAString"
attr(,"package")
[1] "Biostrings"
```

lo que significa que existen componentes o posiciones en la secuencia del genoma dentro cromosoma 10 que están enmascaradas, es decir, que no se muestran. Si nos fijamos en la frecuencia de las letras de nuestro alfabeto, observamos cuáles son las letras enmascaradas y en qué cantidad se encuentran en el genoma

```
> af10_1<-alphabetFrequency(crom10) # frecuencias enmascaradas
> af10_1

       A        C        G        T        M        R
13792137  7928689  7888654 13747168        0        0
       W        S        Y        K        V        H
       0        0        0        0        0        0
       D        B        N        -        +
       0        0        0        0        0

> crom10<-unmasked(crom10) # secuencia desenmascarada, con gaps (huecos)
> class(crom10) # objeto de la clase ‘‘DNAString’’
[1] "DNAString"
attr(,"package")
[1] "Biostrings"

> af10_2<-alphabetFrequency(crom10) # frecuencias desenmascaradas
> af10_2 # N son 110913
```

```
       A        C        G        T        M        R
13792137  7928689  7888654 13747168        0        0
       W        S        Y        K        V        H
       0        0        0        0        0        0
       D        B        N        -        +
       0        0   110913        0        0
```

Como ya indicamos anteriormente, las posiciones etiquetadas con una "N" son nucleótidos del genoma todavía desconocidos o sin determinar. Para realizar el análisis, y poder calcular las matrices de transición y de emisión necesarias que son nuestros parámetros, tenemos que trabajar con la secuencia desenmascarada. Decido eliminar los "gaps" etiquetados con "N" para calcular las frecuencias del resto de nucleótidos y dinucleótidos (secuencias de dos nucleótidos de longitud), y poder calcular ambas matrices. Con ellas, aplicaremos el algoritmo de Viterbi.

Calculo la matriz de transición a la que llamaré "trans", de dimensión 8×8 de acuerdo a los 8 estados (A_+, C_+, G_+, T_+, A_-, C_-, G_-, T_-; en este orden), de la siguiente forma: calculo las tablas de contingencia de todos los posibles dinucleótidos en las regiones de islas y de océano, por lo que obtengo dos matrices 4×4. En efecto, utilizando las frecuencias de los dinucleótidos, puedo conocer las transiciones de un estado o nucleótido a otro. Dimensiono estas matrices en los bloques diagonales de mi matriz "trans". Los restantes espacios de la matriz de transición miden las probabilidades de transición de una isla a un océano (bloque superior) y al revés (bloque inferior). Para calcularlas, tomo la primera letra antes de una isla y la primera que está después de una isla, para todas las islas de "posic10"; y calculo las frecuencias (o transiciones) con la primera de la isla y la última de la isla, respectivamente. Dimensiono el resultado en dos matrices 4×4 que añado de nuevo a mi matriz "trans" en el orden correspondiente.

```
> # Modelo de islas

> cadenamas<-matrix(NA,3889,1) # cadena de islas con gaps
> for (i in 1:3889)
+ cadenamas[i,]<-as.character(crom10[(start[i]+1):(end[i]-1)])

> # juntamos filas de ‘‘cadenamas’’
> juntomas<-do.call(paste,c(as.list(cadenamas),sep=""))
> juntomas<-as(juntomas,"DNAString") # objeto ‘‘character’’ en ‘‘DNAString’’

> cantidad_n<-countPattern("N",juntomas) # elimino los gaps
> m_n<-matchPattern("N",juntomas)
> start2<-start(m_n)
> juntomas[start2]<-NULL

> # dinucleótidos
```

```
> vmas<-strsplit(gsub("([[:alnum:]]2)", "\\ 1", juntomas), " ")[[1]]
> contingmas<-table(vmas) # tabla de contingencia para el modelo de islas
> contingmas<-matrix(contingmas,4,4,dimnames = list(c("A","C","G","T"),
+ c("A","C","G","T")))
> contingmas<-t(contingmas)
> contingmas

      A     C     G     T
A 40922 33374 34410 31473
C 37691 32438 44137 33764
G 36865 44800 33534 33423
T 24626 36763 37682 40345
```

Realizo los mismos cálculos para el modelo de océano teniendo en cuenta la siguiente matriz

```
> # Modelo de océano

> cadenamenos<-matrix(NA,3890,1) #con gaps
> cadenamenos[1,]<-as.character(crom10[1:startmenos[1]])
> for (i in 1:3888)
+ cadenamenos[i+1,]<-as.character(crom10[endmenos[i]:startmenos[i+1]])
> cadenamenos[3890,]<-as.character(crom10[endmenos[3889]:43467561])
```

y obtengo la tabla de contingencia para el modelo "–"

```
> contingmenos

        A       C       G       T
A 2372255 1195707 1197175 1988388
C 1548899  714664  345723 1202780
G 1100241  802326  712796 1179329
T 1733899 1101888 1537068 2361162
```

Unimos ahora ambas matrices en "trans"

```
> trans<-matrix(0,8,8,dimnames =
+ list(c("A+","C+","G+","T+","A-","C-","G-","T-"),
+ c("A+","C+","G+","T+","A-","C-","G-","T-")))
> trans[1:4,1:4]<-contingmas
> trans[5:8,5:8]<-contingmenos
> trans

      A+    C+    G+    T+      A-      C-      G-      T-
A+ 40922 33374 34410 31473       0       0       0       0
C+ 37691 32438 44137 33764       0       0       0       0
```

```
G+ 36865 44800 33534 33423       0       0       0       0
T+ 24626 36763 37682 40345       0       0       0       0
A-     0     0     0     0 2372255 1195707 1197175 1988388
C-     0     0     0     0 1548899  714664  345723 1202780
G-     0     0     0     0 1100241  802326  712796 1179329
T-     0     0     0     0 1733899 1101888 1537068 2361162
```

y rellenamos los bloques restantes como sigue

```
> trans3<-matrix(NA,dim(posic10)[1],2)
> for (i in 1:dim(posic10)[1])
+ trans3[i,1]<-as.character(crom10[(start[i]-1):start[i]]) # océano a isla
+ trans3[i,2]<-as.character(crom10[end[i]:(end[i]+1)]) # isla a océano
+ }

> inicial<-table(trans3[,1]) # transiciones de océano a isla
> inicim<-matrix(0,4,4,dimnames = list(c("A-","C-","G-","T-"),
+ c("A+","C+","G+","T+")))
> inicim[,2]=c(inicial[1],inicial[2],inicial[3],inicial[4])

> final<-table(trans3[,2]) # transiciones de océano a isla, elimino Ns
> final<-matrix(final[-c(8,14)],4,4,dimnames =
+ list(c("A-","C-","G-","T-"), c("A+","C+","G+","T+")))
> final<-t(final)

> trans[1:4,5:8]<-final
> trans[5:8,1:4]<-inicim
> trans
```

```
      A+    C+    G+    T+      A-      C-      G-      T-
A+ 40922 33374 34410 31473     266     142     207     250
C+ 37691 32438 44137 33764     409     281      43     363
G+ 36865 44800 33534 33423     212     175     216     296
T+ 24626 36763 37682 40345     220     223     294     290
A-     0  1063     0     0 2372255 1195707 1197175 1988388
C-     0   826     0     0 1548899  714664  345723 1202780
G-     0  1155     0     0 1100241  802326  712796 1179329
T-     0   845     0     0 1733899 1101888 1537068 2361162
```

```
> trans<-prop.table(trans,1) # matriz de transición final
> trans
```

```
          A+           C+        G+        T+          A-
A+ 0.2901364 0.2366211962 0.2439664 0.2231431 0.001885936
C+ 0.2527460 0.2175207543 0.2959712 0.2264126 0.002742647
```

```
G+ 0.2465540 0.2996234643 0.2242762 0.2235338 0.001417861
T+ 0.1753452 0.2617645593 0.2683081 0.2872696 0.001566472
A- 0.0000000 0.0001573745 0.0000000 0.0000000 0.351206469
C- 0.0000000 0.0002166335 0.0000000 0.0000000 0.406226822
G- 0.0000000 0.0003042799 0.0000000 0.0000000 0.289853885
T- 0.0000000 0.0001254666 0.0000000 0.0000000 0.257451303
            C-           G-          T-
A+ 0.001006778 0.0014676271 0.001772497
C+ 0.001884313 0.0002883468 0.002434183
G+ 0.001170404 0.0014446131 0.001979655
T+ 0.001587833 0.0020933760 0.002064895
A- 0.177021456 0.1772387894 0.294375912
C- 0.187433581 0.0906721197 0.315450844
G- 0.211369426 0.1877831219 0.310689288
T- 0.163609588 0.2282256117 0.350588030
```

```
> rowSums(trans) # las filas suman 1
```

```
A+ C+ G+ T+ A- C- G- T-
 1  1  1  1  1  1  1  1
```

Observamos en nuestra matriz de transición que las probabilidades son más bajas en las matrices o bloques no diagonales y altas de los bloques diagonales. Esto se debe a que, en general, es más probable quedarse en una isla o en una región de océano que moverse de un modelo a otro. Además, las transiciones de océano a isla son cero excepto en la segunda columna, ya que una isla siempre comienza por el nucleótido C. Por otro lado, en el genoma hay menos cantidad de regiones de islas que de océano, por lo tanto, las probabilidades de transición del modelo "+" al modelo "−" deberán ser mayores.

Calculamos a continuación la matriz de emisiones como hemos indicado en la teoría.

```
> emision
```

```
    A C G T
A_+ 1 0 0 0
C_+ 0 1 0 0
G_+ 0 0 1 0
T_+ 0 0 0 1
A_- 1 0 0 0
C_- 0 1 0 0
G_- 0 0 1 0
T_- 0 0 0 1
```

```
> rowSums(emision) # es también una matriz estocástica

A_+ C_+ G_+ T_+ A_- C_- G_- T_-
  1   1   1   1   1   1   1   1
```

Una vez que hemos calculamos ambas matrices, la de transiciones y la de las probabilidades de emisión, pasamos a aplicar el algoritmo de Viterbi para encontrar las regiones de islas CpG en nuestro ejemplo. El paquete "HMM" de R tiene ya implementado tal algoritmo, que nos sirve para comprobar los resultados de nuestra programación.

En primer lugar, se eliminan los "gaps" del cromosoma 10 como anteriormente, y se calcula la distribución de probabilidad inicial, que tiene 8 valores para los 8 estados.

```
> for (i in 1:3889)
+ nucle[i,]<-letterFrequency(crom10[start[i]:end[i]],
+ letters=c("A","C","G","T")) # no cuento N
> }

> nucle<-colSums(nucle) # frecuencias absolutas en islas, más C y G
[1] 281148 300391 299284 279448

> nucle_tot<-c(af10_2[1],af10_2[2],af10_2[3],af10_2[4])
> nuclem<-nucle_tot-nucle # frecuencias absolutas en océanos, menos C y G
[1] 13510989  7628298  7589370 13467720

> prob_inicial<-c(nucle,nuclem)/sum(nucle_tot) # distribución inicial
> prob_inicial
[1] 0.006484542 0.006928372 0.006902840 0.006445332
    0.311624390 0.175942983 0.175045128 0.310626412

> sum(prob_inicial) # suma 1
[1] 1
```

Implementamos el algoritmo de Viterbi aplicando la teoría estudiada. Utilizaremos logaritmos para evitar que las probabilidades se anulen (multiplicaciones consecutivas de números cada vez más pequeños, tendentes a cero). El objeto "crom10" es ahora una cadena de caracteres, donde cada caracter es un nucleótido. Se modifica de esta forma para poder aplicar a "crom10" la función "viterbi" del paquete de R "HMM", y así comprobar nuestros resultados. Además, se han eliminado de tal secuencia los posibles valores "N".

```
> crom10<-strsplit(gsub("([[:alnum:]]1)",
+ "\\ 1", crom10), " ")[[1]]
> seq<-crom10[start[2707]:end[2707]] # isla con longitud mínima 23
```

```
> x<-c("A","C","G","T") # símbolos u observaciones
> pi<-c("A_+","C_+","G_+","T_+","A_-","C_-","G_-","T_-") # estados
> n=length(seq)
> delta_1<-matrix(NA,8,n)
> delta_2<-matrix(NA,8,n)
> oculta<-numeric(n)
> prob_oculta<-numeric(n)

> # PASO i=1 # paso de inicialización
> for (i in 1:8)
+ delta_1[i,1]<-log(prob_inicial[i]*emision[i,as.character(seq[1])])
+ delta_2[i,1]<-0
+ }

> # PASO i=2,3,...,n # paso de inducción
> for (i in 2:n)
+ for (j in 1:8)
+ maxi<-NULL
+ for (k in 1:8)
+ temp<-max(delta_1[k,i-1]+log(trans[k,j]))
+ delta_2[j,i]<-which.max(delta_1[,i-1]+log(trans[,j]))
+ maxi<-max(maxi,temp)
+ }
+ delta_1[j,i]<-maxi+log(emision[j,as.character(seq[i])])
+ }
+ }
```

Con el procedimiento "hacia atrás" obtenemos el camino oculto final

```
> chi=numeric(n)
> chi[n]<-which.max(delta_1[,n])
> oculta[n]<-pi[chi[n]]
> prob_oculta[n]<-max(delta_1[,n]) # probabilidad de la secuencia oculta

> for (i in n:2)
+ chi[i-1]<-delta_2[chi[i],i]
+ oculta[i-1]<-pi[chi[i-1]]
+ prob_oculta[i-1]<-max(delta_1[,i]+log(trans[,chi[i]]))
+ }

> chi # secuencia de los valores que dan el máximo

 [1] 2 3 2 3 1 2 3 4 4 3 4 3 4 4 3 2 4 1 1 2 2 3 4

> oculta # secuencia oculta que corresponde a isla
```

```
 [1] "C_+" "G_+" "C_+" "G_+" "A_+" "C_+" "G_+" "T_+" "T_+"
[10] "G_+" "T_+" "G_+" "T_+" "T_+" "G_+" "C_+" "T_+" "A_+"
[19] "A_+" "C_+" "C_+" "G_+" "T_+"
```

Comprobamos ahora que la secuencia obtenida es la misma que la que se obtiene aplicando la función "viterbi" del paquete de R "HMM".

```
> library(HMM)
> HMM<-initHMM(pi,x,prob_inicial,trans,emision)
> VITERBI<-viterbi(HMM,crom10[start[2]:end[2]])
> oculta==VITERBI
```

```
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[12] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[23] TRUE
```

De esta forma, hemos conseguido detectar y predecir las zonas de islas CpG presentes en el cromosoma 10 del pez cebra. Si probásemos con más datos (no sólo de islas), como por ejemplo una parte grande del cromosoma (entero requeriría más tiempo de ejecución), obtendríamos regiones de islas y océanos aproximadamente coincidentes con el artículo mencionado, en el que se basan nuestros datos de entrenamiento. Éste era nuestro objetivo, y hemos comprobado también su veracidad gracias al paquete "HMM" de R utilizado.

5.5. Otras aplicaciones

5.5.1. Modelos de Markov ocultos en Bioinformática: estudio de las proteínas

A la hora de conocer las funciones de las proteínas, es de vital importancia saber de qué forman "pliegan" estas proteínas en su estructura tridimensional. Además, si queremos predecir los cambios en las funciones de las mismas, podemos hacerlo utilizando los modelos de Markov ocultos basándonos en los cambios de los aminoácidos que las componen ([25]).

El ratio en el que los polimorfismos de un solo nucleótido no sinónimos (SNPns, del inglés *Single Nucleotide Polymorphism No Synonyms*), que son aquellas variaciones que dentro de una secuencia codificante pueden modificar la cadena de aminoácidos que producen, se están identificando en el genoma humano está aumentando considerablemente debido a los avances en la secuenciación del genoma y exoma (parte del genoma compuesta por exones) enteros. Por ello, los métodos automáticos capaces de distinguir entre SNPns patogénicos y funcionales están asumiendo gran importancia. Los modelos de Markov ocultos se utilizan en este campo para describir el análisis funcional (FATHMM, del inglés *Functional Analysis through*

Hidden Markov Models) y predecir los efectos funcionales de las variaciones en las proteínas. Utilizando un modelo de peso para las mutaciones humanas, obtenemos aproximaciones que superan a los métodos tradicionales de predicción. FATHMM puede aplicarse eficientemente a una gran cantidad de proyectos de secuenciación del genoma humano y no humano, con el beneficio añadido de las asociaciones de resultados fenotípicos.

5.5.2. Modelos de Markov ocultos en filogenia molecular

Otro de los estudios publicados recientemente que utilizan modelos de Markov ocultos es el de las relaciones evolutivas (filogenética) y las especiaciones entre los humanos, gorilas, y chimpancés. Se trata de estimar en qué momento se ha producido la "separación" genética de las especies ([6]).

La relación genealógica entre los humanos, los chimpancés y los gorilas varía a lo largo del genoma. Se desarrolla un modelo de Markov oculto que incorpora esta variación y relaciona los parámetros del modelo con cantidades de población genética, como los tiempos de especiación (paso de la población de una especie a otra u otras) y el tamaño de las poblaciones ancestrales. Así, en las simulaciones no se observa sesgo entre las estimaciones del modelo de Markov oculto. Se aplica el modelo a cuatro alineamientos autosómicos contiguos: humano - chimpancé - gorila - orangután comparando un total de 1.9 millones de pares de bases. Encontramos un muy reciente tiempo de especiación del hombre al chimpancé (4.1 ± 0.4 millones de años), y tamaños efectivos de población ancestral bastante grandes ($65,000 \pm 30,000$ para el antepasado del humano - chimpancé - gorila).

Por tanto, se afirma que que compartimos un ancestro común con el chimpancé y que la diferenciación de las dos especies fue hace 4 millones de años.

Bibliografía

[1] Alex Sánchez. *Cadenas de Markov y aplicaciones en biología computacional*. Departament d'Estadística U.B. Estadística i Bioinformática. http://www.ub.edu/stat/docencia/EADB/EstadBioinformatica_Web/ /materials/Apunts

[2] Alex Sánchez. *Introducción a los Modelos de Markov Ocultos*. Departament d'Estadística U.B. Estadística i Bioinformática. http://www.ub.edu/stat/docencia/EADB/EstadBioinformatica_Web/ /materials/Apunts

[3] Alex Sánchez. *Modelos de Markov Ocultos. Predicción de genes*. Departament d'Estadística U.B. Estadística i Bioinformática. http://www.ub.edu/stat/docencia/EADB/EstadBioinformatica_Web/ /materials/Apunts

[4] Anders Krogh (1998) *An Introduction to Hidden Markov Models for Biological Sequences*. Computational Methods in Molecular Biology, pages 45-63.

[5] Anders Krogh, I. Saira Mian, David Haussler (1994). *A hidden Markov model that finds genes in E. coli DNA*. Oxford University Press. Nucleic Acids Research, Vol. 22, No 22.

[6] Asger Hobolth, Ole F. Christensen, Thomas Mailund, Mikkel H. Schierup (Febrero 2007). *Genomic Relationships and Speciation Times of Human, Chimpanzee, and Gorilla Inferred from a Coalescent Hidden Markov Model*. PLoS Genet 3 (2): e7.

[7] Barbara Resch. *Hidden Markov Models*. http://www.igi.tugraz.at/lehre/CI/tutorials/HMM.zip

[8] R. BARTOSZYNSKI, M. NIEWIADOMSKABUGAJ (2008).*Probabiliy and Statistical Inference*. Wiley. New York. 13, 491-494.

[9] C. Burge, S. Karlin (1997). *Prediction of complete gene structures in human genomic DNA*. Journal of Molecular Biology.

[10] Byung-Jun Yoon (2009). *Hidden Markov Models and their Applications in Biological Sequence Analysis*. Bentham Sciencie Publishersn Ltd.

[11] Christopher Nemeth (2011). *Hidden Markov Models with Applications to DNA Sequence*.

[12] David Kulp, David Haussler (Baskin Center of Computer Engineering and Information Sciences, University of California); Martin G. Reese, Frank H. Eeckman (Genome Informatics Group, Lawrence Berkeley National Laboratory). *A Generalized Hidden Markov Models for the Recognition of Human Genes in DNA*.

[13] R. Durbin, S. Eddy, A. Krogh, G. Mitchison (1998) *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press. http://www.cs.ubbcluj.ro/ csatol/mach_learn/bemutato/ /Mate_Korosi_HMMpres.pdf

[14] Durret R. (1999). *Essentials of stochastic processes*. Springer, New York, USA.

[15] Enlace de bioinformática básica, secuenciación del ADN. http://ocw.unia.es/ciencias-de-la-vida/bioinformatica-basica/materiales/bloque-4/04-1-Sequencing-1-v2_Lectura %20del %20ADN-%20de %20los %20organismos.pdf

[16] Enlace sobre genética básica. http://recursostic.educacion.es/secundaria/edad/4esobiologia/4quincena7/ /pdf/quincena7.pdf

[17] Enlace sobre cadenas de Markov. http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/PEst/ /tema4pe.pdf

[18] Enlaces sobre contrastes de hipótesis. http://fcm.ens.uabc.mx/ chelo/estadistica/doc-pdf/lec-6-4-5.pdf http://pendientedemigracion.ucm.es/info/ecocuan/anc/ectriaqf/noparam.pdf

[19] Enlace sobre ejercicios con Cadenas de Markov. http://www.few.vu.nl/ wvanwie/Courses/BiomedicalMathematics/ /Exercise_Lecture1_MarkovChain__0112012.pdf

[20] Enlace sobre la estructura de los genes eucariotas. http://www.cs.tav.ac.il

[21] Gardiner-Garden M., Frommer M. (1987). *CpG islands in vertebrate genomes*. J. Mol. Biol. 196 (2): 261-82.

[22] Gary A. Churchill (1989). *Stochastic Models for Heterogeneous DNA Sequences*. Pergamon Press plc, Society for Mathematical Biology. Bulletin of Mathematical Biology Vol. 51, No 1, pp. 79-94.

[23] Guy Leonard Kouemou. *History and Theoretical Basics of Hidden Markov Models*. EADS Deutschland GmbH, Germany.

[24] Hao Wu, Brian Caffo, Harris A. Jaffee, Rafael A. IRIZARRY (2010). *Redefining CpG islands using hidden Markov models*. Biostatistics, **11**, 3, pp. 499-514.

[25] Hashem A. Shihab et al. (October 2012).*Predicting the Functional, Molecular, and Phenotypic Consequences of Amino Acid Substitutions using Hidden Markov Models*. Publicado online en: http://www.ncbi.nlm.nih.gov/pubmed/23033316.

[26] Harmen J. Bussemaker, Hao Li, Eric D. Siggia (2000). *Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis*. PNAS vol. 97, No. 18, 10096-10100.

[27] Jose L. Oliver et al. (2001). *Isochore chromosome maps of eukaryotic genomes*. Gene 276, 47?56.

[28] B. H. Juang, L. R. Rabiner (1993). *Fundamentals of Speech Recognition*. Prentice Hall.

[29] Kevin P. Murphy (2002). *Hidden semi-Markov models (HSMMs)*. http://www.ai.mit.edu/ murphyk

[30] Kulkarni V. G. (1995). *Modeling and Analysis of Stochastic Systems*. Chapman Hall, London, UK. http://www.bioconductor.org/help/course-materials/2002/Wshop/lect3.pdf

[31] Lawrence R. Rabiner (1989). *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, Vol. 77, No. 2.

[32] Lior Pachter.*Applications of hidden Markov models to sequence analysis*.

[33] Máthé Zoltán, Korosi Zoltán (2006). *Hidden Markov Models in Bioinformatics*.

[34] Nello Cristianini, Matthew W. Hahn (2006). *Introduction to Computational Genomics. Hidden Markov models*. Cambridge University Press. 4, 60-67.

[35] Richard Hughey, Anders Krogh (1995). *Hidden Markov models for sequence analysis: extension and analysis of the basic method*. Computer Engineering, University of California, Santa Cruz.

[36] Richard Cowan. *Expected frequencies of DNA patterns using Whittle's formula*. Departament of Statistics, University of Hong Kong.

[37] Roberto A. Pava. *Búsqueda de repeticiones en secuencias de ADN*. Universidad Nacional de Colombia.

[38] Sean R. Eddy (2004). *What is a hidden Markov model ?*. Nature Publishing Group.

[39] Verónica Saladrigas, Gonzalo Claros (2002). *Vocabulario Inglés-Español de Bioquímica y Biología Molecular (1.ª entrega)*. Panace@, Boletín de Medicina y Traducción 2002; 3 (9-10) 13-28.
http://www.medtrad.org/Panacea/PanaceaPDFs/Diciembre2002.htm

[40] Warren. J. Ewens, Gregory Grant (2005). *Statistical Methods in Bioinformatics. An Introduction*. Springer.