Detección de puntos de cambio en secuencias de ADN mitocondrial

Nora Martínez Villanueva

Máster en Técnicas Estadísticas

Universidade de Vigo

Detección de puntos de cambio en secuencias de ADN mitocondrial

Nora Martínez Villanueva

Autorización de entrega

D. Javier Roca Pardiñas y D. Miguel Mendoça Fonseca

Certifican

Que el proyecto titulado "Detección de puntos de cambio en secuencias de ADN mitocondrial" ha sido realizado por Dña. Nora Martínez Villanueva, con D.N.I. 53179846-M, bajo la dirección de D. Javier Roca Pardiñas y D. Miguel Mendoça Fonseca.

Esta memoria constituye la documentación que, con nuestra autorización, entrega dicho alumno como Proyecto Fin de Máster.

Firmado

Javier Roca Pardiñas Miguel M. Fonseca

Vigo, a 16 de Enero de 2012

Resumen

Identificar los procesos de mutación que afectan a las secuencias de ADN es fundamental para una mejor comprensión de como evolucionan los genomas. El mecanismo de replicación, durante el cual las cadenas se exponen a un elevado daño mutacional, se ha descrito como una de las principales fuentes de sesgo en la composición nucleotídica las cadenas. En este trabajo se presenta `seq2R`, un paquete de `R` que detecta singularidades en la composición de genomas mitocondriales (ADNmt). Para ello, se han implementado técnicas de suavización tipo *kernel* que estiman los índices nucleotídicos y se han aplicado métodos *bootstrap* en la construcción de intervalos de confianza para dichas estimaciones. Además, este paquete permite representar gráficamente la estimaciones obtenidas y realiza inferencia sobre los puntos de cambio (o singularidades) de interés.

Índice general

Capítulo 1

Introducción

La mayoría de los organismos eucariotas contienen dentro de sus células unos orgánulos que se conocen con el nombre de mitocondrias. Dichos orgánulos son esenciales para la actividad celular ya que son los responsables de convertir las calorías que incorporamos en la dieta en energía utilizable (adenosin trifosfato, ATP) a través del proceso de fosforilación oxidativa (Wallace, 1992). Sin embargo, dicho proceso no es el único en el que intervienen las mitocondrias. Por ejemplo, se sabe que están implicadas en la biosíntesis de otros metabolitos celulares, y en la regulación de la muerte celular programada o apoptosis (Orrenius, 2004).

Estos orgánulos se componen de una membrana mitocondrial externa, espacio intermembranoso, membrana mitocondrial interna (con invaginaciones denominadas crestas) y matriz mitocondrial. Aunque la mayor parte del ADN de una célula está en el núcleo, la mitocondria tiene su propio genoma, el ADN mitocondrial (ADNmt, Fig. 1.1)(Bruces et al., 2007).

El número de mitocondrias por célula varía ampliamente según el tipo de organismo o tejido y se estima que cada una de ellas tenga de 2-10 copias de ADNmt (Wiesner et al., 1992).

El genoma de las mitocondrias se localiza en la matriz mitocondrial, y tiene una estructura típicamente circular constituida por dos cadenas de ADN. Éstas se componen principalmente de cuatro bases nitrogenadas: adenina (a), timina (t), guanina (g) y citosina (c). La unión de ambas cadenas se produce por el apareamiento de dichas bases, la adenina y la timina son complementarias, mientras que la guanina lo es con la citosina. Por su composición bioquímica, las dos hebras son diferentes, ya que la secuencia nucleotídica de una es rica en G (cadena pesada o $H-strand$) y la otra cadena es pobre en este base nitrogenada (cadena ligera o $L-strand$) (Anderson

et al., 1981). El genoma mitocondrial codifica 13 proteínas implicadas en la cadena respiratoria, 2 ARNs ribosómicos, y 22 ARNs tranferentes, los cuales están asociados con el proceso de transcripción de ADNmt (P. F. Chinnery, 2003). En la Figura 1.2 se representa un esquema del ADNmt humano.

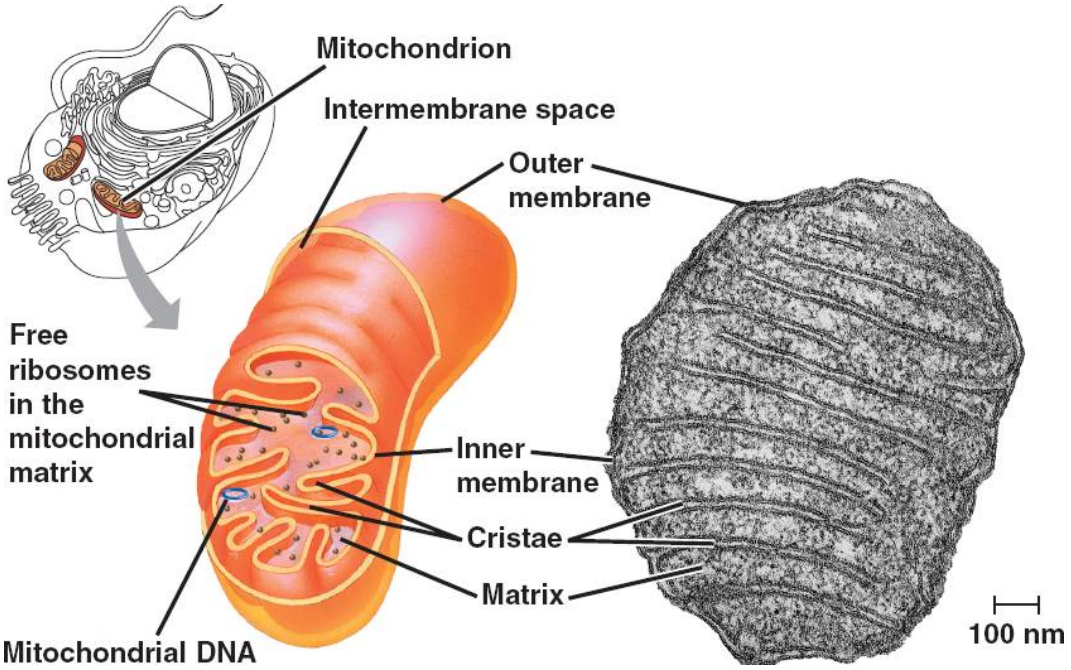

Fig. 1.1: Estructura de una mitocondria (centro) de una célula eucariótica (izq.). Imagen al miscroscopio de una mitocondria (dcha.) (http://bio1151b.nicerweb.com/Locked/media/ch06/mitochondrion.html).

El descubrimiento de este genoma único en las mitocondrias fue un paso muy importante para poder realizar estudios sobre el origen y evolución de dichos orgánulos (Mounolou et al., 1966; Schatz, 1963).

Una mutación es un cambio de un nucleótido por otro. La variación genética en el ADNmt se origina a través de mutaciones que se acumulan en el genoma. La tasa de mutación promedio del ADNmt es 10 veces mayor que la del ADN nuclear. Esto es debido a que (i) el ADNmt está expuesto al daño oxidativo causado por las reacciones que se producen en la mitocondria, (ii) el ADN nuclear está mejor protegido y (iii) los mecanismos de reparación de daños del ADN son poco eficientes en las mitocondrias. Dado que el ADNmt se hereda por vía materna (Dawid and Blackler, 1972; 3rd Hutchison et al., 1974) y la tasa de recombinación es limitada y

rara vez genera nuevas variantes genéticas (Tsaousis et al., 2005), dichas mutaciones son mayoritariamente la fuente de variación en este genoma.

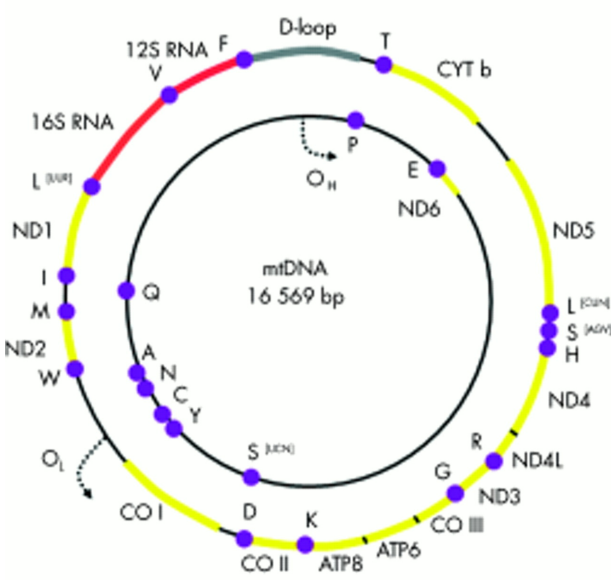

Fig. 1.2: Genoma mitocondrial *Homo sapiens*. Es una pequeña molécula de 16569 kb de ADNmt bicatenaria que codifica 13 componentes esenciales en la cadena respiratoria: ND1-ND6 genes que codifican 7 subunidades del complejo I, el Cyt *b* codifica la subunidad del complejo III, CO I-III codifican tres subunidades del complejo IV, los genes ATP6 y ATP8 codifican para dos subunidades del complejo V. Además, contiene 2 genes de ARN ribosómico (12S y 16 S ARNr), y 22 genes de ARN transferente. D-loop es una región no codificante implicada en la regulación de procesos importantes, OH y OL, son los orígenes de replicación de la cadena pesada y de la cadena ligera del ADNmt. Abreviaturas: ND1 a ND6, subunidades 1-6 de NADH deshidrogenasa; Cyt b, subunidad de citocromo *b*; COI-III, subunidades de citocromo *c* oxidasa; ATP6 y ATP8, subunidades de ATP sintasa. 12S y 16S ARNr; genes del ARNt se indican con una letra del amino ácido correspondiente.

Cuando los mecanismos mutagénicos y el proceso de selección afectan por igual a ambas cadenas de ADN, la frecuencia nucleotídica en cada una ellas debería estar equilibrada, segunda regla de la paridad (Chargaff, 1950; Lobry, 1995). Sin embargo, el sesgo en la composición de las cadenas se puede identificar como desviaciones en esta relación, lo que implica la existencia de mutaciones asimétricas derivadas de distintos mecanismos de mutación, por ejemplo, cambios de bases durante la replicación, transcripción o reparación del ADN (Frank and Lobry, 1999). Si estas mutaciones tienen lugar durante la replicación cabe esperar que aparezcan grandes cambios en la

composición nucleotídica en el origen de replicación (en animales vertebrados presentan dos, denominados OH y OL) y en el *terminus* de las nuevas secuencias de ADNmt (Touchon and Rocha, 2008).

Basándose en la composición de las secuencias y con el fin de estimar la ubicación de los dos orígenes de replicación (OH y OL), Grigoriev (1998) utilizó el sesgo acumulado GC: un método que consiste en la suma de (G-C) / (G + C) desde un punto de inicio arbitrario de la secuencia hasta recorrerla por completo. Pudo observar que el sesgo GC aumenta cuando nos acercamos a OH y OL. Sin embargo, este método al igual que otros muchos utilizados hasta el momento, carece de rigor estadístico.

En este proyecto se presenta una nueva metodología estadística que permite detectar cambios en la composición de las secuencias genómicas mediante modelos de regresión. Determinar estos puntos de cambio resultan de gran ayuda para comprender la evolución de los genomas en distintos organismos.

Un modelo de regresión describe la relación entre una variable explicativa o covariable X y una variable respuesta Y. En un contexto no paramétrico, la relación entre X e Y puede explicarse como

$$Y = m(X) + \varepsilon \tag{1.1}$$

donde m es una función suave, y ε es el error que se asume independiente de la covariable X.

El modelo en (1.1) podría aplicarse a diferentes conjuntos de datos procedentes de campos científicos muy dispares, como en este caso la genética evolutiva, o de manera más general, la bioinformática. Actualmente, y debido a los avances computacionales alcanzados en las últimas décadas, ambas disciplinas están en auge.

En estas áreas resulta de gran interés el estudio del genoma mitocondrial. La comunidad científica trata de comprender la evolución del ADNmt, así como su mantenimiento, ya que se ha descubierto que mutaciones en esta molécula pueden causar enfermedades en humanos. Estas mutaciones en las secuencias de ADNmt dan lugar a cambios en la composición nucleotídica.

En la metodología estadística que se describe en el Capítulo 2 se propone el uso de suavizadores locales lineales tipo *kernel* (Wand and Jones, 1995). La ventaja de estos estimadores no paramétricos es que dan lugar a curvas flexibles y de fácil interpretación. Para hacer inferencia sobre los puntos de cambio, y finalmente extraer conclusiones, es imprescindible la construcción de intervalos de confianza. Además, en este proyecto se propone el uso de técnicas de remuestreo *bootstrap* (Efron, 1979; Efron and Tibshirani, 1993). Adicionalmente, debido a la enorme cantidad de datos con la que se suele trabajar en bioinformática (ADNmt humano 16569 pb), se ha implementado la técnica de aceleración computacional *binning* (Fan and Marron, 1994).

Fuera del contexto matemático-estadístico, muchos usuarios pueden estar interesados en utilizar esta metodología. Por lo que un objetivo fundamental en este proyecto es la implementación de un software sencillo y amigable con la metodología desarrollada, la librería `seq2R`.

El proyecto se estructura en 4 capítulos diferenciados. En el Capítulo 2 se explica la metodología utilizada, como por ejemplo la estimación de los puntos de cambio, los intervalos de confianza *bootstrap*, la técnica *binning*, etc. El desarrollo de software, incluidas las funciones programadas hasta el momento, se describen en el Capítulo 3. El Capítulo 4 se centra en la aplicación a datos reales, donde se muestran algunos resultados y conclusiones del estudio.

Capítulo 2

Metodología estadística

Cualquier tipo de secuencia de ADN consiste en una larga "frase" formada principalmente por cuatro letras ordenadas (a, t, c, g). Para llevar a cabo un análisis de esta secuencia, es necesario transformarla en cuatro variables (A, T, C y G).

Se define A como una variable binaria donde el valor 0 indica la ausencia de adeninas (a) en una posición determinada X de la secuencia y el 1 su presencia. Las tres restantes variables pueden ser obtenidas de manera análoga.

Para analizar la composición nucleotídica, y detectar así las posibles asimetrías en la secuencia, en este proyecto se propone el uso del *skew profile* (o perfil del sesgo) para A *vs.* T y para C *vs.* G. Estos índices miden las desviaciones de la cantidad de un nucleótido frente a otro y se calculan, para una X dada, de la siguiente manera:

$$AT = (A - T)/(A + T) \quad \text{y} \quad CG = (C - G)/(C + G)$$

Según la segunda regla de la paridad (Chargaff, 1950), el porcentaje de adeninas (a) debe ser aproximadamente igual al porcentaje de timinas (t) y el porcentaje de citosinas (c) similar al de guaninas (g), para cada hebra o cadena de ADN. Usando nuestra notación, esta idea se corresponde con $\sum_{i=1}^{n} A_i \approx \sum_{i=1}^{n} T_i$ y $\sum_{i=1}^{n} C_i \approx \sum_{i=1}^{n} G_i$, siendo n el número de nucleótidos en la secuencia objeto de estudio. En el caso de que una mutación afecte a esta secuencia, esta relación entre pares de nucleótidos se verá alterada.

El índice propuesto permitirá conocer esta relación y tomará un valor próximo a cero en ausencia de mutación. En el caso de un cambio brusco, por ejemplo, en la cantidad de adeninas, el valor del índice AT aumentará considerablemente, lo que se verá reflejado en la curva estimada. Atendiendo a esto, el uso de las derivadas resulta de gran ayuda en este contexto, en concreto, para estimar el punto donde la primera

derivada de AT es máxima o mínima, que se corresponderá con un punto crítico en la tendencia del índice estimado. Este punto reflejará un cambio o desviación en la relación de los dos nucleótidos.

Por simplicidad y a modo de ejemplo se desarrollará la metodología para estimar el índice AT, ya que CG se obtiene del mismo modo. Para estimar dicho índice es necesario primero conocer $\hat{A}$ y $\hat{T}$. Para ello, resulta razonable plantear un modelo de regresión no paramétrica de respuesta binaria como se propone a continuación.

Sea $p_A(X) = \mathrm{P}(A = 1/X = x)$ y $p_T(X) = \mathrm{P}(T = 1/X = x)$, entonces

$$p_A(X) = \frac{exp(m(X))}{1 + exp(m(X))} \quad \text{y} \quad p_T(X) = \frac{exp(m(X))}{1 + exp(m(X))} \tag{2.1}$$

donde m es una función desconocida o el efecto asociado a la covariable X.

2.1. Algoritmo de estimación

De manera general, para estimar en modelo en (2.1) se propone el siguiente algoritmo.

Dada una muestra $\{(X_i, Y_i)\}_{i=1}^n$ los pasos del algoritmo *local scoring* se muestran a continuación.

Inicializa Calcular las estimaciones iniciales, $\hat{m} = log(\bar{Y}/(1 - \bar{Y}))$ y $\hat{p}_i^0 = \hat{p}(X_i) = \bar{Y} = n^{-1} \sum_{i=1}^n Y_i$ $(i = 1, \ldots, n)$.

Paso 1. Calcular las variables dependientes ajustadas $\tilde{Y} = (\tilde{Y}_1, \ldots, \tilde{Y}_n)$ y los pesos $W = (W_1, \ldots, W_n)$,

$$\tilde{Y}_i = \hat{m}(X_i) + \frac{(Y_i - \hat{p}_i^0)}{\hat{p}_i^0(1 - \hat{p}_i^0)} \quad \text{y} \quad W_i = \hat{p}_i^0(1 - \hat{p}_i^0)$$

Las estimaciones de $m(x)$ y de su primera derivada $m^1(x)$ en una posición x se definen como

$$\hat{m}\,(x) = \hat{\beta}_0(x) \quad \text{y} \quad \hat{m}^1\,(x) = \hat{\beta}_1(x) \tag{2.2}$$

donde $\hat{\beta} = \left(\hat{\beta}_0, \hat{\beta}_1\right)$ es el minimizador de

$$\sum_{i=1}^{n} \left(\tilde{Y}_i - \beta_0 - \beta_1\,(X_i - x)\right)^2 h^{-1} K\left(\frac{X_i - x}{h}\right) W_i$$

donde $K(u) = 1/\sqrt{2\pi}\exp(-u^2/2)$ es la función Gaussiana tipo *kernel*, y $h > 0$ es el parámetro de suavización (o ventana) y se calculará según el criterio de validación cruzada explicado en la Sección 2.2.

Paso 2. Repetir el **Paso 1** reemplazando p_i^0 por

$$\hat{p}_i = \frac{\exp(\hat{m}(X)_i)}{1 + exp(\hat{m}(X)_i)}$$

para $i = 1, \ldots, n$, hasta que $|D(\hat{\mathrm{p}}^0, \mathrm{Y}) - D(\hat{\mathrm{p}}, \mathrm{Y})|/D(\hat{\mathrm{p}}^0,\ \mathrm{Y}) \leq \varepsilon$, donde ε es un valor muy pequeño y $D(\hat{\mathrm{p}}, \mathrm{Y}) = -2\sum_{i=1}^{n}[Y_i \log(\hat{p}_i) + (1 - Y_i)\log(1 - \hat{p}_i)]$.

Una vez obtenidas las estimaciones de $\hat{A}$ y $\hat{T}$ del modelo en (2.1), es posible calcular el indice correspondiente AT.

$$\widehat{AT} = (\hat{A} - \hat{T})/(\hat{A} + \hat{T})$$

Además, en nuestro estudio, resultará necesario determinar en que puntos o zonas de la secuencia la primera derivada de AT es máxima o mínima. Estos puntos vendrán dados por el maximizador o minimizador de $\widehat{AT}^1$. Sin embargo, en la práctica, ni AT ni AT^1 se conocen, así que el punto crítico buscado debe obtenerse de las estimaciones de $\widehat{AT}$ y $\widehat{AT}^1$ de las verdaderas curvas de AT y AT^1.

Un estimador natural para el punto crítico buscado puede ser definido como el maximizador o minimizador de

$$\widehat{AT}^1(z_1), \ldots, \widehat{AT}^1(z_N)$$

donde $z_1, \ldots, z_N$ es una rejilla o *grid* muy fino de N puntos equidistantes en el rango de los valores de X.

2.2. Selección de la ventana

La implementación del estimador local lineal tipo *kernel* en (2.2) requiere de un proceso de selección del parámetro de suavización o ventana h. Se sabe que las estimaciones no paramétricas obtenidas, basadas en estimación tipo *kernel*, dependen fuertemente de h. El resultado de utilizar una ventana grande es una sobresuavización de la curva, mientras que la elección de una ventana pequeña tiende a reproducir los datos. La selección de la ventana óptima sigue siendo un problema desafiante. Además, hay que tener en cuenta que no existen evidencias que sugieran que la

ventana óptima para estimar m necesariamente deba coincidir con la ventana para estimar su primera derivada m^1. Como solución práctica, en este trabajo, el parámetro de suavización h se selecciona automáticamente minimizando el siguiente criterio de error de validación cruzada (Stone, 1977).

$$VC = \sum_{i=1}^{n} \left(\tilde{Y}_i - \hat{m}^{(-i)} \left(X_i \right) \right)^2 W_i \tag{2.3}$$

donde $\hat{m}^{(-i)}(X_i)$ indica la estimación en X_i, dejando fuera el i-ésimo elemento de la muestra.

2.3. Aspectos computacionales

El método de validación cruzada supone un elevado coste computacional, así como la técnica de remuestreo *bootstrap* (Sección 2.4). Por ello, es fundamental recurrir a alguna técnica de aceleración de cálculo para asegurar que el problema pueda ser abordado de manera adecuada en situaciones prácticas, como la técnica *binning* (Fan and Marron, 1994). El éxito de la técnica *binning* se basa en reducir el número de evaluaciones *kernel*, reemplazando el conjunto de datos $\{(X_i, Y_i)\}_{i=}^{n}$ por otro conjunto reducido sobre el que se realizan las estimaciones. El *binning* lineal se basa en crear un *grid* de N puntos equidistantes a lo largo del rango de X, y asignar a cada punto un peso igual al número de observaciones que hay en su nodo.

Sea $X_1^\bullet < X_2^\bullet < \ldots < X_N^\bullet$ un *grid* de N puntos equidistantes a lo largo del rango de X, con ζ la distancia entre los puntos consecutivos del *grid*. El peso de la i-ésima observación es asignado a los puntos del *grid* más cercanos de acuerdo a

$$W_i r^\bullet = \left(1 - \left|X_i - X_r^\bullet\right| / \zeta\right)_+, r = 1, \ldots, N.$$

De esta manera, la respuesta *binning* $\tilde{Y}_r$ y los pesos *binnig* $\tilde{W}_r$ para $r = 1, \ldots, N$ se construyen como se muestra a continuación:

$$W_r^\bullet = \sum_{i=1}^{n} W_i^{\bullet r} \text{ y } Y_r^\bullet = \frac{1}{W_r^\bullet} \sum_{i=1}^{n} W_i^{\bullet r} \tilde{Y}_i,$$

y la aproximación *binning* del estimador $\hat{m}$ en (2.2) se obtiene minimizando

$$\sum_{r=1}^{N} \left(Y_r^\bullet - \beta_0 - \beta_1 \left(X_r^\bullet - x\right)\right)^2 h^{-1} K \left(\frac{X_i^\bullet - x}{h} \right) W_r^\bullet,$$

Como en el proceso de estimación, la técnica *binning* puede ser aplicada al error de validación cruzada obteniéndose

$$VC \approx \sum_{r=1}^{N} W_r^{\bullet} \left(\frac{Y_r^{\bullet(-r)}}{W_r^{\bullet}} - \hat{m}^{(-r)} \left(X_r^{\bullet} \right) \right)^2$$

La elección del número de puntos del *grid* es un compromiso entre el error de aproximación y la velocidad computacional: cuanto más fino sea el *grid* de puntos seleccionados mejor serán las aproximaciones *binning*. En este trabajo se ha seleccionado un N= 400 puntos a lo largo de el rango de X, que se consideró suficiente. Sin embargo, dependiendo del tamaño de muestra n y de la distribución de la covariable, puede ser apropiado utilizar un mayor número de puntos en el *grid*.

2.4. Intervalos de confianza

Para hacer inferencia sobre las curvas curvas estimadas en la Sección 2.1 o sobre los puntos críticos obtenidos es imprescindible llevar a cabo la construcción de los intervalos de confianza. Para realizar esta tarea, resulta necesario conocer la distribución de las estimaciones anteriores. Sin embargo, es sabido que, en un contexto de regresión no paramétrica, la teoría asintótica que determina esos percentiles no está cerrada, y el uso de las técnicas de remuestreo *bootstrap* introducidas por Efron (1979) (ver también Efron and Tibshirani, 1993; Härdle and Mammen, 1993; Kauermann and Opsomer, 2003) parecen una buena alternativa.

Los métodos *bootstrap* son métodos de remuestreo para analizar la variabilidad del las estimaciones obtenidas de la muestra original. Dada la naturaleza de los datos, el método seleccionado ha sido el *bootstrap* binario. Los pasos para construir los intervalos de confianza para un valor $\widehat{AT}$ obtenido del modelo en (2.1) son los siguientes:

Paso 1. Se obtiene la estimación de $\widehat{AT}$ de la muestra original y las estimaciones piloto de las medias condicionadas

$$\hat{p}_A(X_1), \ldots, \hat{p}_A(X_n) \quad \text{y} \quad \hat{p}_T(X_1), \ldots, \hat{p}_T(X_n)$$

Paso 2. Para $b = 1, \ldots, B$ (p.ej. B=1000), se generan muestras *bootstrap* $\left\{(X_i, A_i^{\bullet b})\right\}_{i=1}^{n}$ y $\left\{(X_i, T_i^{\bullet b})\right\}_{i=1}^{n}$ con

$$A_i^{\bullet b} \sim \text{Bernoulli}(\hat{p}_A(X_i)) \quad \text{y} \quad T_i^{\bullet b} \sim \text{Bernoulli}(\hat{p}_T(X_i))$$

y se calculan la correspondiente estimación de $\widehat{AT}^{\bullet p}$.

Finalmente, el intervalo de confianza al $100(1-\alpha)\,\%$ de AT viene dado por

$$I = \left(\widehat{AT}^{\alpha/2}, \widehat{AT}^{1-\alpha/2}\right)$$

donde $\widehat{AT}^{p}$ representa el $p-$percentil de los valores de $\widehat{AT}^{\bullet 1}, \ldots, \widehat{AT}^{\bullet B}$.

Capítulo 3

Desarrollo de software

Hasta el momento se ha discutido sobre el tipo de datos y la metodología estadística aplicada a los mismos. A partir de ahora se describirán las funciones implementadas en el paquete `seq2R` de `R` (R Development Core Team, 2009) . Este software proporciona salidas numéricas y gráficas de los modelos de regresión no paramétrica revisados en el Capítulo 2.

La ventaja de `R` respecto a otros lenguajes de programación estadísticos, como puede ser Fortran, FORmula TRANslation (Fortran 95 Language Guide, 1995), es la sencillez, que permite a usuarios no expertos en este campo hacer uso práctico de la metodología implementada. Sin embargo, `R` cuenta con una desventaja bien conocida: su elevado coste computacional. Por ello, se ha desarrollado la librería `seq2R` cuyas funciones implementadas en `R` han sido programadas en Fortran. Esta librería presenta dependencias del paquete `seqinr`.

El nombre de la librería `seq2R` hace referencia a la abreviatura de "Sequence to R". Esta última letra engloba dos conceptos: (1) programa R, (2) Recuperar. Ambos conceptos derivan de que esta librería permite, además de cargar ficheros .fasta[1] o .gbk[1], Recuperar secuencias de la base de datos GenBank[2] y llevar a cabo análisis su análisis con R.

En este capítulo se presenta el paquete `seq2R` en detalle con datos de ADN mitocondrial humano y se ha estructurado de la siguiente manera. En la Sección 3.1 se describe la función `read.genbank()` que permite recuperar de la base de datos GenBank secuencias de ADNmt con el fin de analizarlas posteriormente. La función `read.all()` se muestra en la Sección 3.2 y su uso permite al usuario leer secuencias

[1]Este tipo de ficheros basados en texto, son muy utilizados en bioinformática para representar secuencias de ADN, las bases se representan usando códigos de una letra.

[2]Base de datos creada en E.E.U.U. Actualmente, está gestionada por NCBI.

de nucleótidos con formato .fasta o .gbk. Con la función `change.binary()` los cuatro nucleótidos se convierten a código binario de ceros y unos (Sección 3.3). La función principal del paquete es `change.points()`, útil para detectar cambios en la composición nucleotídica de los genomas (Sección 3.4). Para representar las salidas de la función anterior, el usuario dispone de la función `plot.change.points()` (Sección 3.5). Por último, en la Sección 3.6, se describe la función `critical()` cuyo fin es detectar puntos críticos en la secuencia objeto de análisis.

3.1. Función `read.genbank()`

En determinadas situaciones el usuario carece del archivo de datos con la secuencia que se pretende analizar. En este contexto, le resultaría de gran ayuda aplicar la función `read.genbank()`. Esta función utiliza el siguiente enlace, http://eutils.ncbi.nlm.nih.gov/entrez/eutils/ (Bethesda, 2006) a partir del cual recupera las secuencias para su análisis futuro. El único argumento que necesita `read.genbank()` es el locus o número de acceso de la secuencia (Tabla 3.1). La llamada de la función se muestra a continuación:

```
read.genbank(locus)
```

Argumento	Descripción
`locus`	Vector de caracteres con el locus o número de acceso de la secuencia. Valores perdidos no están permitidos.

Tabla 3.1: Argumentos de la función `read.genbank`.

La sintaxis específica para un ejemplo de ADNmt humano se muestra a continuación. En este ejemplo, se pueden observar algunas de las 16569 letras que conforman dicho genoma, el código de la secuencia y el nombre científico del organismo objeto de estudio.

```
R> library(seq2R)
R> humanDNA = read.genbank("NC_012920")
```

```
R> humanDNA
   [[1]]
   [1] "g" "a" "t" "c" "a" "c" "a" "g" "g" "t" "c" "t"
   [13] "a" "t" "c" "a" "c" "c" "c" "t" "a" "t" "t" "a"
   [25] "a" "c" "c" "a" "c" "t" "c" "a" "c" "g" "g" "g"
   ...
   [16537] "c" "c" "a" "c" "a" "c" "g" "t" "t" "c" "c" "c"
   [16549] "c" "t" "t" "a" "a" "a" "t" "a" "a" "g" "a" "c"
   [16561] "a" "t" "c" "a" "c" "g" "a" "t" "g"

   [[2]]
   [1] "NC_012920"

   attr(,"species")
   [1] "Homo_sapiens"
```

3.2. Función `read.all()`

En la sección anterior se ha descrito la función que permite al usuario cargar secuencias en `R` vía internet. Sin embargo, en ciertas ocasiones el usuario cuenta con las secuencias incluidas en archivos. Esta necesidad ha impulsado el desarrollo de una nueva función, `read.all()`, cuya característica principal es la lectura de dos tipos de ficheros ampliamente utilizados en bioinformática, `.fasta` o `.gbk`.

Los argumentos de la función se describen en la Tabla 3.2. La llamada de la función se muestra a continuación:

```
read.all(file, seqtype= "DNA")
```

Argumento	Descripción
`file`	Vector de caracteres con el nombre del fihero.
`seqtype`	Vector de caracteres para el tipo de secuencia (ADN por defecto)

Tabla 3.2: Argumentos de la función `read.all`.

Por simplicidad y a modo de ejemplo, se ha incluido la sintaxis de un tipo de fichero que contiene la secuencia de ADNmt humano (`ADNmthum.gbk`).

```
R> library(seq2R)
R> humanDNA = read.all("ADNmthum.gbk")
R> humanDNA
   [[1]]
   [1] "g" "a" "t" "c" "a" "c" "a" "g" "g" "t" "c" "t"
   [13] "a" "t" "c" "a" "c" "c" "c" "t" "a" "t" "t" "a"
   [25] "a" "c" "c" "a" "c" "t" "c" "a" "c" "g" "g" "g"
   [37] "a" "g" "c" "t" "c" "t" "c" "c" "a" "t" "g" "c"
   ...
   [16525] "a" "g" "c" "c" "t" "a" "a" "a" "t" "a" "g" "c"
   [16537] "c" "c" "a" "c" "a" "c" "g" "t" "t" "c" "c" "c"
   [16549] "c" "t" "t" "a" "a" "a" "t" "a" "a" "g" "a" "c"
   [16561] "a" "t" "c" "a" "c" "g" "a" "t" "g"

   [[2]]
   [1] "NC_012920 16569 bp"
```

3.3. Función `change.binary()`

La función `change.binary()` convierte la secuencia biológica al sistema binario para facilitar los cálculos numéricos en la estimación del modelo (Tabla 3.3).

Argumento	Descripción
`x`	Objeto de la clase `read.genbank` o `read.all`.

Tabla 3.3: Argumentos de la función `change.binary`.

La sintaxis para el ADNmt humano se muestra a continuación. Se puede observar que la función devuelve una lista con dos componentes `$AT` y `$CG`. Para cada componente, la variable X indica la posición de las bases nucleotídicas en la secuencia genómica. En el caso de las cuatro variables binarias A, T, C, G se representa con un 0 la ausencia del nucleótido en una posición dada y con un 1 la presencia del mismo. Se han separado las bases en estos dos componentes (AT y CG) ya que, según la bibliografía, no todas las bases nitrogenadas son igualmente sensibles al daño mutagénico;

la citosina o guanina presentan una mayor sensibilidad que la adenina o incluso la timina.

```
R> humanDNAbin=change.binary(humanDNA)
R> humanDNAbin
   $AT
            X A T
   [1,]     2 1 0
   [2,]     3 0 1
   [3,]     5 1 0
   [4,]     7 1 0
   [5,]    10 0 1
   ...

   $CG
            X C G
   [1,]     1 0 1
   [2,]     4 1 0
   [3,]     6 1 0
   [4,]     8 0 1
   [5,]     9 0 1
   ...
```

3.4. Función `change.points()`

La función principal del paquete es `change.points()` que permite crear un objeto de clase `change.points`. La función `change.points()` ajusta un modelo de regresión no paramétrica mediante suavizadores locales lineales tipo *kernel*, para posteriormente, calcular el perfil del sesgo con las estimaciones obtenidas. El modelo tiene como variable explicativa la posición de los nucleótidos en la secuencia, mientras que la respuesta es la variable binaria correspondiente a un nucleótido (A, T, C o G) obtenida con la función `change.binary()`, descrita anteriormente.

El perfil del sesgo o *skew profile* de AT se obtiene aplicando $(\hat{A} - \hat{T})/(\hat{A} + \hat{T})$, siendo $\hat{A}$ y $\hat{T}$ las estimaciones de A y T obtenidas anteriormente. De igual modo se obtiene el perfil correspondiente a CG.

Los argumentos de la función se describen en la Tabla 3.4. La llamada de la función se muestra a continuación:

```
change.points(x, kbin=400, p=1, h=NULL, W=1, nboot=200)
```

Argumento	Descripción
`x`	Objeto de la clase `change.binary`.
`kbin`	Número de nodos binnig.
`p`	Grado del polinomio.
`h`	Ventana o parámetro de suavización.
`W`	Vector con los pesos.
`nboot`	Número de repeticiones *bootstrap*.

Tabla 3.4: Argumentos de la función `change.points`.

La función `print.change.points()` devuelve un breve resumen numérico con algunos resultados del ajuste del modelo: número de nucleótidos de $A+T$ y $C+G$, número de nodos *binning*, número de repeticiones *bootstrap* y la ventana o parámetro de suavización. Por último, a través de un argumento lógico, `TRUE` o `FALSE`, indica la presencia o ausencia de al menos algún punto crítico.

```
R> hDNA=change.points(humanDNAbin,kbin=400,nboot=1000)
R> hDNA
Call:
change.points(x = humanDNAbin, kbin = 400, nboot = 1000)

Number of A+T base pairs:9218

Number of G+C base pairs:7350

Number of binning nodes: 400

Number of bootstrap repeats: 1000

Banwidth: 89.3

Exists any critical point? TRUE
```

3.5. Función plot.change.points()

La función `plot.change.points()` permite representar gráficamente las estimaciones del *skew profile*, así como su primera derivada, tanto para *AT* como para *CG*. Además, se incluyen los intervalos de confianza *bootstrap*. Las salidas gráficas dependerán de los argumentos que se incluyan en la función `plot.change.points()` (Tabla 3.5).

Argumento	Descripción
`x`	Objeto de la clase `change.points`.
`base.pairs`	Cadena de caracteres para el *skew profile* de "AT" y/o "CG".
`der`	Número que determina qué curva se dibuja en el gráfico. Si `der = 0` se muestra la estimación del *skew profile*. Si `der = 1` en el gráfico se representa su primera derivada.
`xlab`	Título para el eje de abcisas.
`ylab`	Título para el eje de ordenadas.
`col`	Color para la estimación y primera derivada.
`ICcol`	Color para los intervalos de confianza (estimación y primera derivada).
`main`	Título principal del gráfico.
`type`	Tipo de gráfico que se desea dibujar.
`ICtype`	Tipo de gráfico que se desea dibujar para los intervalos de confianza.

Tabla 3.5: Argumentos de la función `plot.change.points`.

El resultado del siguiente código se muestra en la Fig. 3.1.

```
plot.change.points(hDNA, base.pairs="AT")
```

3.6. Función critical()

La última de las funciones implementadas hasta el momento en el paquete es `critical()`. La característica principal de esta función consiste en determinar los valores de la variable X (posiciones en la secuencia), con sus respectivos intervalos

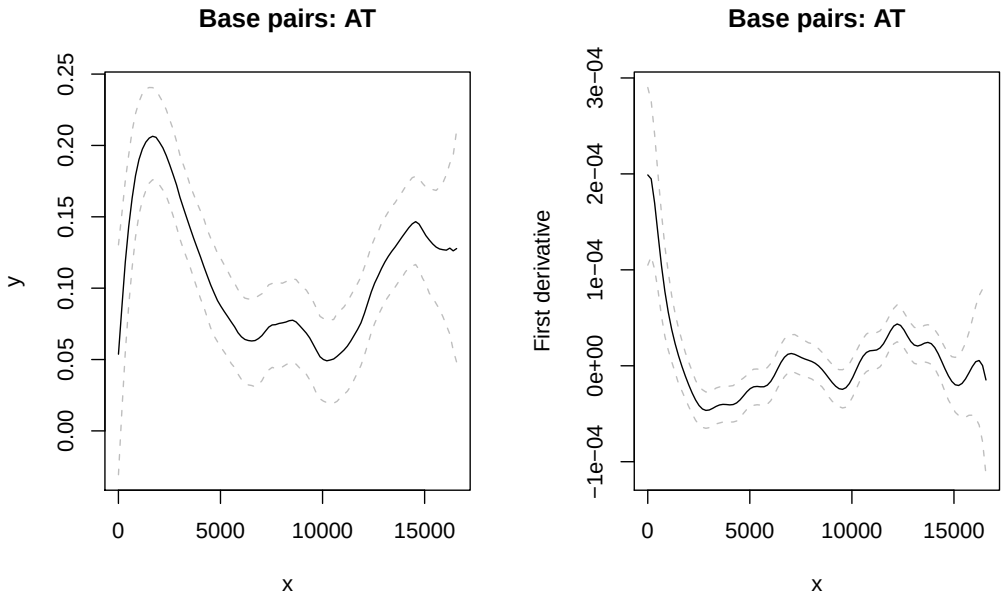

Fig. 3.1: Datos procedentes del ADNmt humano. Panel izq.: estimación del *skew profile A vs. T*. Panel Dcha. Primera derivada del *skew profile A vs. T*. Línea gris discontinua: intervalos de confianza *bootstrap* 95 %.

de confianza *bootstrap* al 95 %, donde la primera derivada de la estimación del *skew profile* alcanza un máximo o un mínimo (puntos críticos).

Los argumentos de la función se describen en la Tabla 3.6.

Argumento	Descripción
`x`	Objeto de la clase `change.points`.
`base.pairs`	Vector de caracteres para "AT" o "CG".

Tabla 3.6: Argumentos de la función `plot.change.points`.

La siguiente sintaxis muestra un ejemplo de aplicación de `critical()`.

```
R> critical(hDNA)
$AT
      Critical 95% low_CI 95% up_CI
[1,]  2949.83    1953.38  4277.40
[2,]  9675.88    9260.69  9924.99
[3,] 12250.04   11295.11 13371.05
[4,] 13744.72   13454.09 13786.24
[5,] 14865.73   14865.73 14907.25
```

```
$CG
       Critical 95% low_CI  95% up_CI
[1,]   665.38   333.19   665.38
[2,]  3281.38  2575.48  3904.24
[3,]  5648.23  5233.00  6146.52
[4,]  8305.76  7724.43  9260.81
[5,] 10008.23  9842.14 10132.81
[6,] 10589.57 10465.00 10672.62
[7,] 11918.33 11586.14 12250.52
[8,] 13081.00 12914.90 13122.52
[9,] 13662.33 13662.33 13786.90
[10,] 16402.90 16319.86 16527.48
```

Capítulo 4

Estudio del ADN mitocondrial en *Homo sapiens*

Durante años, la comunidad científica estuvo convencida de que la replicación del ADNmt de animales vertebrados ocurría asimétricamente, a través de dos orígenes de replicación, OH y OL (Shadel and Clayton, 1997) (Fig. 4.1). Sin embargo, varios estudios cuestionan este modelo de replicación asimétrica(Holt et al., 2000; Yasukawa et al., 2005) y a cambio proponen que la síntesis de las nuevas cadenas del ADNmt ocurre de forma clásica-sincronizada entre ambas hebras(Reyes et al., 2005) (Fig. 4.1). Aunque los investigadores todavía no han llegado a una uniformidad en las opiniones sobre el proceso de replicación en vertebrados, lo que sí parece razonable es que existen otros orígenes de replicación adicionales a OL y a OH (Brown et al., 2005).

En esta Sección se describe el análisis de la composición del ADN mitocondrial humano mediante la librería `seq2R`. La secuencia objeto de estudio procede de la base de datos GenBank (número de locus/acceso `NC_012902`). Con dicho análisis se pretende detectar los orígenes de replicación y, poder acercarnos un poco más a la respuesta de cómo ocurre el proceso de replicación en sí mismo.

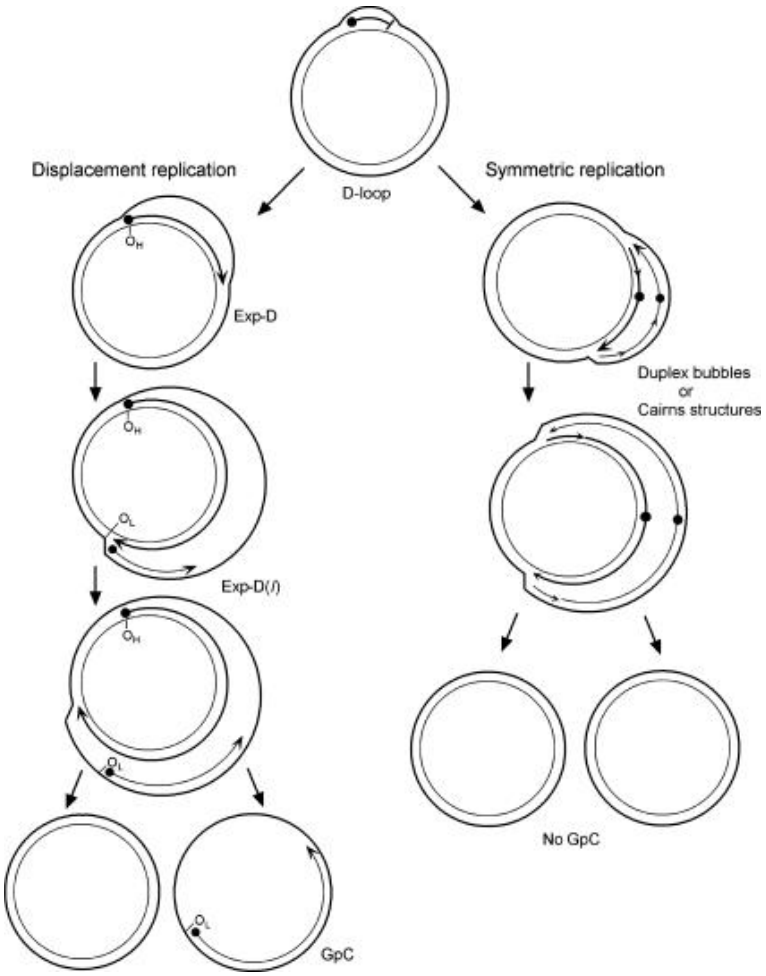

Fig. 4.1: Modelos de replicación asimétrica y simétrica.(Brown et al., 2005)

En la Fig. 4.2 se representa el *skew profile* para A *vs.* T, tanto la estimación como su primera derivada junto con sus intervalos de confianza al 95 %. En el panel superior se pueden observar en color rojo y azul los puntos de cambio encontrados, que se corresponden con mínimos y máximos de la primera derivada, respectivamente (panel inferior). Los correspondientes valores se pueden observar en la Tabla 4.1. De la misma manera, en la Fig. 4.3, se muestra el análisis para C *vs.* G. En este caso los valores críticos se indican en la Tabla 4.2.

Puntos críticos	**IC 95 %**
2949.83	(1953.38, 4277.40)
7184.75	(7101.71, 7309.31)
9675.88	(9260.69, 9924.99)
12250.04	(11295.11, 13371.05)
13744.72	(13454.09, 13786.24)
14865.73	(14865.73, 14907.25)

Tabla 4.1: Puntos críticos para ***A vs. T*** con sus intervalos de confianza al 95 %.

Puntos críticos	**IC 95 %**
665.38	(333.19, 665.38)
3281.38	(2575.48, 3904.24)
5648.23	(5233.00, 6146.52)
8305.76	(7724.43, 9260.81)
10008.23	(9842.14, 10132.81)
10589.57	(10465.00, 10672.62)
11918.33	(11586.14, 12250.52)
13081.00	(12914.90, 13122.52)
13662.33	(13662.33, 13786.90)
16402.90	(16319.86, 16527.48)

Tabla 4.2: Puntos críticos para ***C vs. G*** con sus intervalos de confianza al 95 %.

Se sabe que algunos procesos moleculares como la transcripción, recombinación, replicación y reparación pueden afectar a la composición nucleotídica de los genomas mitocondriales. Sin embargo, diversos estudios bioquímicos y evolutivos sugieren que, de los mecanismos anteriores, la replicación juega un papel importante en la composición del ADNmt en organismos vertebrados. Teniendo en cuenta este hecho, y según los análisis previos (Fig. 4.2 y Fig. 4.3), cabe esperar que dichos orígenes se correspondan con puntos de inflexión de la curva *skew profile* donde la pendiente sea negativa y su primera derivada alcance un mínimo local.

A partir del análisis de la composición del ADNmt humano se pueden hacer las siguientes observaciones. Los dos orígenes de replicación, OH y OL, se localizan en regiones donde se han identificado los puntos críticos. En primer lugar, se sabe que OL se localiza en una pequeña región situada entre las posiciones 5730 y 5760. Uno de los puntos críticos identificados en el análisis de *C vs. G* está exactamente localizado en la misma región (Fig. 4.3). Este hecho sugiere que OL es un origen de replicación importante en los genomas mitocondriales de *H. Sapiens*, y a su vez corrobora estudios

bioquímicos recientes que proponen que OL debería tener un papel crucial en el proceso de replicación (Fusté et al., 2010). En segundo lugar, el origen de replicación OH, localizado al comienzo (1-576) y en el *terminus* (16024-16569) de la secuencia, también se ha identificado en el análisis *C vs. G* (Fig. 4.3). Dado que OH es una región fundamental para todas las posibles formas de replicación del ADNmt propuestas hasta la fecha, se puede decir que los resultados de este estudio apoyan que OH es el principal origen de replicación (Brown et al., 2005).

En cuanto al análisis de *A vs. T* (Fig. 4.2) parece que se pueden detectar orígenes de replicación alternativos: (i) en la región corriente arriba o *upstream* de OL, es decir, antes de la posición 5000, (ii) alrededor de las posiciones 10000 y (iii) cercano a la posición 15000. Curiosamente, éste último se ha identificado visualmente utilizando la técnica de microscopía con fuerza atómica (Brown et al., 2005). Sin embargo, los dos primeros no se han descrito anteriormente, por lo que es conveniente realizar otros análisis moleculares en estas regiones (análisis de las estructuras de tallo y lazo, localización de ARNt).

Si los cambios en la composición nucleotídica reflejan el proceso de replicación, y en base a los resultados obtenidos en este trabajo, se puede extraer la siguiente conclusión. Es posible que pueda existir más de un mecanismo de replicación en el ADN mitocondrial humano, ya que se han encontrado indicios evolutivos de la existencia de más de un mecanismo de replicación potencial. De hecho, algunos experimentos en mamíferos *in vivo* realizados por Pohjoismäki et al. (2010) avalan la existencia de más de un mecanismo de replicación.

Los análisis realizados en este proyecto suponen una pequeña aproximación al estudio de las variaciones en el genoma humano y asumen que la composición del ADN mitocondrial se ve significativamente afectada por su modo de replicación. Cabe destacar que otros factores/procesos también pueden influir en la composición nucleotídica del genoma, tales como la selección o mutaciones relacionadas con la transcripción. Por lo tanto, estos resultados deben ser tomados con precaución y complementarse, a su vez, con futuros análisis para confirmar las ideas presentadas en este proyecto.

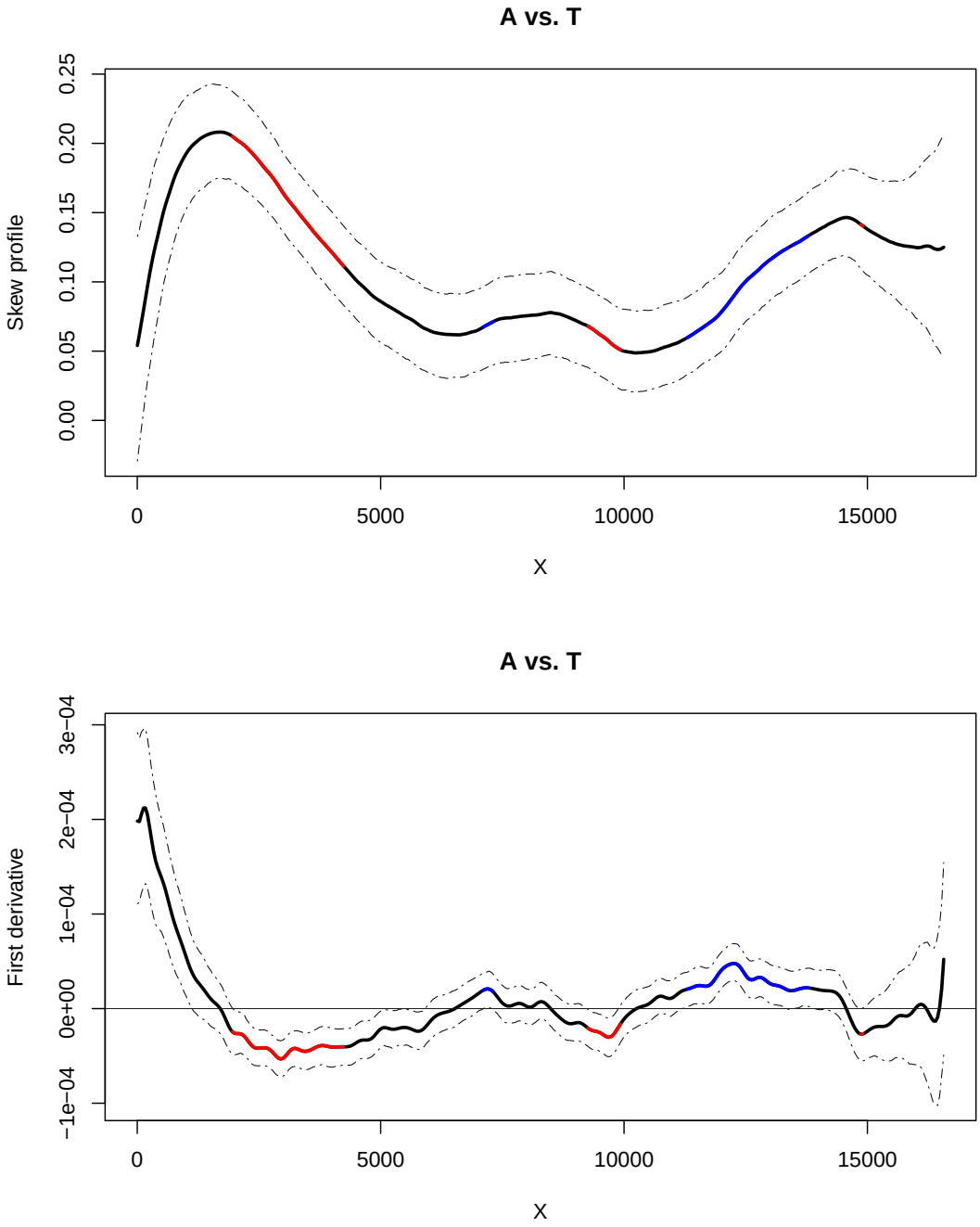

Fig. 4.2: Datos procedentes del ADNmt humano. Panel sup.: estimación del *skew profile A vs. T*. Panel inf.: Primera derivada del *skew profile A vs. T*. Línea discontinua: intervalos de confianza *bootstrap* 95 %. Línea roja: puntos de la variable X que minimizan la primera derivada. Línea azul: puntos de la variable X que maximizan la primera derivada.

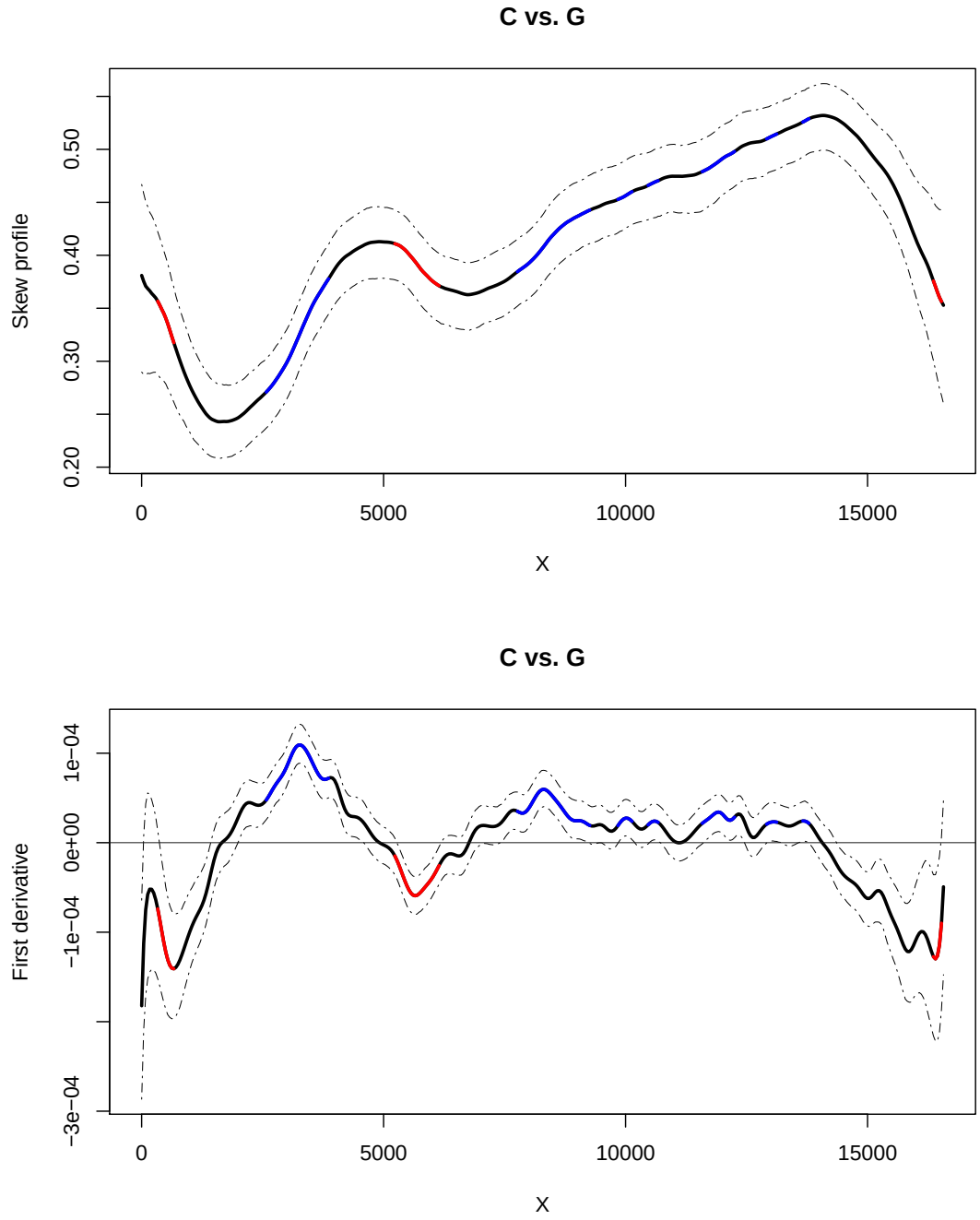

Fig. 4.3: Datos procedentes del ADNmt humano. Panel sup.: estimación del *skew profile C vs. G*. Panel inf.: Primera derivada del *skew profile C vs. G*. Línea discontinua: intervalos de confianza *bootstrap* 95 %. Línea roja: puntos de la variable X que minimizan la primera derivada. Línea azul: puntos de la variable X que maximizan la primera derivada.

Bibliografía

Anderson, S., Bankier, A., Barrel, B., de Bruin, M., Coulson, A., J.Drouin, Eperon, I., Nierlich, D., Roe, B., Sanger, F., Schreier, P., Smith, A., Staden, R., Young, I., 1981. Sequence and organization of the human mitochondrial genome. Nature 290, 457–65.

Bethesda, M.D., 2006. Entrez programming utilities. National Library of Medicine.

Brown, T.A., Cecconi, C., Tkachuk, A.N., Bustamante, C., Clayton, D.A., 2005. Replication of mitochondrial dna occurs by strand displacement with alternative light-strand origins, not via a strand-coupled mechanism. Genes & Development 19, 2466–2476.

Bruces, A., Alexander, J., Julian, L., Martin, R., Keith, R., Peter, W., 2007. Molecular Biology of the Cell. Garland Science. 4th edition.

Chargaff, E., 1950. Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. Experentia 6, 201–9.

Dawid, I.B., Blackler, A.W., 1972. Maternal and cytoplasmic inheritance of mitochondrial dna in xenopus. Developmental Biology 29, 152 – 161.

Efron, B., 1979. Bootstrap methods: another look at the jackknife. Annals of Statistics 7, 1–26.

Efron, E., Tibshirani, R.J., 1993. An introduction to the Bootstrap. Chapman and Hall, London.

Fan, J., Marron, J., 1994. Fast implementation of nonparametric curve estimators. Journal of Computational and Graphical Statistics 3, 35–56.

Frank, A., Lobry, J., 1999. Asymmetric substitution patterns: a review of possible underlying mutational or selective mechanisms. Gene 238, 65 – 77.

Fusté, J., Wanrooij, S., Jemt, E., Granycome, C., Cluett, T., Shi, Y., Atanassova, N., Holt, I., Gustafsson, C., Falkenberg, M., 2010. Mitochondrial RNA Polymerase Is Needed for Activation of the Origin of Light-Strand DNA Replication. Mol Cell 37, 67–78.

Gehrke, W., 1995. Fortran 95 Language Guide. Springer, London.

Grigoriev, A., 1998. Analyzing genomes with cumulative skew diagrams. Nucleic Acids Research 26, 2286–2290.

Härdle, W., Mammen, E., 1993. Comparing nonparametric versus parametric regression fits. The Annals of Statistics 21, pp. 1926–1947.

Holt, I.J., Lorimer, H.E., Jacobs, H.T., 2000. Coupled leading- and lagging-strand synthesis of mammalian mitochondrial dna. Cell 100, 515–524.

3rd Hutchison, C.A., Newbold, J.E., Potter, S.S., Edgell, M.H., 1974. Maternal inheritance of mammalian mitochondrial DNA. Nature 5475, 536–8.

Kauermann, G., Opsomer, J., 2003. Local Likelihood Estimation in Generalized Additive Models. Scandinavian Journal of Statistics 30, 317–337.

Lobry, J.R., 1995. Properties of a general model of dna evolution under no-strand-bias conditions. Journal of Molecular Evolution 41, 680.

Mounolou, J.C., Jakob, H., Slonimski, P.P., 1966. Mitochondrial DNA from yeast "petite" mutants: specific changes in buoyant density corresponding to different cytoplasmic mutations. Biochemical and Biophysical Research Communications 2, 218–24.

Orrenius, S., 2004. Mitochondrial regulation of apoptotic cell death. Toxicology Letters 149, 19 – 23.

P.F. Chinnery, E.A.S., 2003. Mitochondria. Journal of Neurology Neurosurgery Psychiatry 9, 1188–99.

Pearson, W.R., Lipman, D.J., 1988. Improved tools for biological sequence comparison. Proceedings of the National Academy of Sciences 85, 2444–2448.

Pohjoismäki, J.L.O., Holmes, J.B., Wood, S.R., Yang, M.Y., Yasukawa, T., Reyes, A., Bailey, L.J., Cluett, T.J., Goffart, S., Willcox, S., 2010. Mammalian mitochondrial

DNA replication intermediates are essentially duplex but contain extensive tracts of RNA/DNA hybrid. Journal of Molecular Biology .

R Development Core Team, 2009. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

Reyes, A., Yang, M.Y., Bowmaker, M., Holt, I.J., 2005. Bidirectional replication initiates at sites throughout the mitochondrial genome of birds. Journal of Biological Chemistry 280, 3242–3250.

Schatz, G., 1963. The isolation of possible mitochondrial precursor structures from aerobically grown baker's yeast. Biochemical and Biophysical Research Communications 12, 448–51.

Shadel, G.S., Clayton, D.A., 1997. Mitochondrial dna maintenance in vertebrates. Annual Review of Biochemistry 66, 409–435.

Stone, C.J., 1977. Consistent nonparametric regression. The Annals of Statistics 5, 595–620.

Touchon, M., Rocha, E.P., 2008. From GC skews to wavelets: A gentle guide to the analysis of compositional asymmetries in genomic data. Biochimie 90, 648 – 659.

Tsaousis, A.D., Martin, D.P., Ladoukakis, E.D., Posada, D., Zouros, E., 2005. Widespread recombination in published animal mtdna sequences. Molecular Biology and Evolution 22, 925–933.

Wallace, D., 1992. Diseases of the mitochondrial dna. Annu Rev Biochem 61.

Wand, M.P., Jones, M.C., 1995. Kernel Smoothing. Chapman & Hall, London.

Wiesner, R.J., Rüegg, J.C., Morano, I., 1992. Counting target molecules by exponential polymerase chain reaction: Copy number of mitochondrial dna in rat tissues. Biochemical and Biophysical Research Communications 183, 553 – 559.

Yasukawa, T., Yang, M., Jacobs, H., Holt, I., 2005. A bidirectional origin of replication maps to the major noncoding region of human mitochondrial dna. Mol Cell 18, 651–62.

Anexo
Package 'seq2R'

Type Package

Title Simple method to detect compositional changes in genomic sequences.

Version 1.0

Date 2012-01-08

Author Nora M.Villanueva and Javier Roca-Pardiñas

Maintainer Nora M. Villanueva <nmvillanueva@uvigo.es>

Description This software is useful for loading .fasta or .gbk files, and for retrieving sequences from GenBank dataset. This package allows to detect differences or asymmetries based on nucleotide composition by using local linear kernel smoothers. Also, it is possible to draw inference about critical points (i. e. maximum or minimum points) related with the derivative curves. Additionally, bootstrap method have been used for estimating confidence intervals and speed computational techniques have been implemented in "seq2R".

License GPL

LazyLoad yes

R topics documented:

`seq2R-package` *Simple method to detect compositional changes in genomic sequences.*

Description

`seq2R` is just a shortcut for "Sequence to `R`". The last letter means two concepts: (i) `R` program and (ii) `R`etrieve. This software is useful for loading .fasta or .gbk files, and for recovering sequences from GenBank dataset. This package allows to detect differences or asymmetries based on nucleotide composition by using local linear kernel smoothers. Also, it is possible to draw inference about critical points (i. e. maximum or minimum points) related with the derivative curves. Additionally, bootstrap method have been used for estimating confidence intervals and speed computational techniques have been implemented in "seq2R".

Details

Package:	seq2R
Type:	Package
Version:	1.0
Date:	2012-01-08

Author(s)

Nora M. Villanueva Javier Roca-Pardiñas.

Maintainer: Nora M. Villanueva <nmvillanueva@uvigo.es>

References

Bethesda, M.D., (2006). Entrez programming utilities. National Library of Medicine. http://eutils.ncbi.nlm.nih.gov/entrez/eutils/.

Gehrke, W., 1995. Fortran 95 Language Guide. Springer, London.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. Annals of Statistics, 7:126.

Efron, E. and Tibshirani, R. J. (1993). An introduction to the Bootstrap. Chapman and Hall, London.

Pearson, W.R., Lipman, D.J., (1988). Improved tools for biological sequence comparison. Proceedings of the National Academy of Sciences 85, 2444:2448.

Touchon, M., Rocha, E.P., (2008). From GC skews to wavelets: A gentle guide to the analysis of compositional asymmetries in genomic data. Biochimie 90, 648:659.

Wand, M.P., Jones, M.C., (1995). Kernel Smoothing. Chapman & Hall, London

---

`read.genbank` *Read DNA sequences from GenBank via internet.*

Description

This function connects to the GenBank database, and reads nucleotide sequences using locus code given as arguments.

Usage

```
read.genbank(locus)
```

Arguments

`locus`	Character string giving by locus code or accession number.

Details

This function uses http://eutils.ncbi.nlm.nih.gov/entrez/eutils/ (E - utilities) from where the sequences are downloaded. E-utilities are a set of eight server-side programs that provide a stable interface into the Entrez query and database system at the National Center for Biotechnology Informatio (NCBI). The E-utilities use a fixed URL syntax that translates a standard set of input parameters into the values necessary for various NCBI software components to search for and retrieve the requested data. The E-utilities are therefore the structured

interface to the Entrez system, which currently includes 38 databases covering a variety of biomedical data, including nucleotide and protein sequences, gene records, three-dimensional molecular structures, and the biomedical literature.

Value

`Sequence` The returned list has a component `"Sequence"` containing the DNA sequence taken from the field "ORIGIN" in GenBank. The sequence is a vector of single characters.

`Locus or accession`
The returned list has a component `"Locus/Accession"` containing the names of the locus or accession number taken from the field "LOCUS" or "ACCESSION" in GenBank.

`Species` The returned list has an attribute `"Species"` containing the names of the species taken from the field "ORGANISM" in GenBank.

Note

If the computer is not connected to the internet, this function will not work.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas.

Examples

```
## load human mitochondrial DNA sequence
library(seq2R)
humanDNA <- read.genbank("NC_012920")
humanDNA
```

`read.all` *read FASTA and GBK formatted files*

Description

Read nucleic acid sequences from a file in FASTA or GBK format.

Usage

```
read.all(file = system.file(""), seqtype = "DNA")
```

Arguments

`file`	The name of the file which the sequences in fasta or gbk format are to be read from. to the current working directory, `getwd`. The default here is to read the `ct.fasta` file which is present in the `sequences` folder of the seqinR package.
`seqtype`	The nature of the sequence: `DNA`

Details

FASTA is a widely used format in molecular biology. Sequence in FASTA format begins with a single-line description (distinguished by a greater-than '>' symbol), followed by sequence data on the next lines. Lines starting by a semicolon ';' are ignored, as in the original FASTA program (Pearson and Lipman 1988). The sequence name is just after the '>' up to the next space ' ' character, trailling infos are ignored for the name but saved in the annotations.

GenBank format is a flat format for sequence data related to complete genomes. By convention, GenBank format files have the extension gbk. Files contain fields with different types of information well-labeled. The header of the file has information describing the sequence, such as its type, shape, length and source. The features of the genome sequence follow the header, and include protein translations. The DNA sequence is the last element of the file, which ends with (and must include) a soluble slash. Complete genomes in this format are available at the ftp://ftp.ncbi.nih.gov/.

Value

`Sequence`	The returned list has a component `"Sequence"` containing the DNA sequence taken from the field "ORIGIN" in GenBank. The sequence is a vector of single characters.
`Locus or accession`	The returned list has a component `"Locus/Accession"` containing the names of the locus or accession number taken from the field "LOCUS" or "ACCESSION" in GenBank. Also, return sequence length.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas

Examples

```
# human mitochondrial DNA data
library(seq2R)
humanDNA <- read.all("ADNmthum.gbk")
humanDNA
```

`change.binary` *Convert biological sequences into binary code.*

Description

Biological sequences are categorical variables. With this function `change.binary` the four nucleotides are coded with two bits, 0 and 1 (binary numeral system) for being used by almost all modern computers.

Usage

```
change.binary(x, ...)
```

Arguments

`x`	Sequences in fasta or gbk format are to be change from. The nature of the sequence is DNA. Sequences are returned as a vector of single characters.

Value

The returned list has two component (`$AT`, `$CG`). Both of them containing a matrix with values about their critical (maximum and minimum) points, lower and upper confidence intervals 95 %.

`AT`	Variable A and T with binary system.
`CG`	Variable C and G with binary system.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas.

Examples

```
# human mitochondrial DNA data
library(seq2R)
humanDNA <- read.genbank("NC_012920")
humanDNAbin <- change.binary(humanDNA)
humanDNAbin
```

`change.points` *Simple method to detect compositional changes in genomic sequences.*

Description

`change.points` is used to detect change at genomic sequence composition. The method is based on fitting nonparametric models by using local linear kernel smoothers.

Usage

```
change.points(x, kbin = 400, p = 1, h=NULL, W = 1, nboot=200,...)
```

Arguments

`x`	Sequences in binary system (by using `change.binary` function previously) are to be analyzed from.
`kbin`	Number of equally spaced points at which to estimate the curves. The number of binning nodes over which the function is to be estimated.
`p`	Degree of a polynomial.
`h`	The kernel bandwidth smoothing parameter for adenine, thymine, guanine and citosine nucleotides. Large values of bandwidth make smoother estimates, smaller values of bandwidth make less smooth estimates.
`W`	Weights on the data.
`nboot`	Number of bootstrap repeats.

Details

For each genomic sequence the AT and CG skews profiles were calculated as $AT = (\hat{A} - \hat{T})/(\hat{A} + \hat{T})$, $CG = (\hat{C} - \hat{G})/(\hat{C} + \hat{G})$. For both skews, the dependent variable (X) was defined by the genome position and the response variable was defined by the skew profile (AT, CG). Additionally, we also calculated the first derivative to analyze the slope variation of the skew values and to detect critical points (maximum or minimum).

Value

The function computes and returns a list of short information for a fitted `change.points` object.

`Number of A+T base pairs`
: The returned value is the total nucleotide (adenine and thymine) contained in the sequence analyzed.

`Number of C+G base pairs`
: The returned value is the sum of citosine and guanine contained at the sequence.

`Number of binning nodes`
: Number over which the function is to be estimated.

`Number of bootstrap repeats`
: Total value of bootstrap used to fit the model.

`Bandwidth` Kernel bandwidth or smoothing parameter.

`Exists any critical point`
: Emphasize if there is or not any critical.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas.

Examples

```
# human mitochondrial DNA data
library(seq2R)
humanDNA <- read.genbank("NC_012920")
humanDNAbin <- change.binary(humanDNA)
hDNA<-change.points(humanDNAbin)
```

`print.change.points`
Short summary for change.points

Usage

```
print.change.points(x, ...)
```

Arguments

`model` `change.points` object.

Value

The function computes and returns a list of short information for a fitted `change.points` object.

`Number of A+T base pairs`
: The returned value is the total nucleotide (adenine and thymine) contained in the sequence analyzed.

`Number of C+G base pairs`
: The returned value is the sum of citosine and guanine contained at the sequence.

`Number of binning nodes`
: Number over which the function is to be estimated.

`Number of bootstrap repeats`
: Total value of bootstrap used to fit the model.

`Bandwidth`
: Kernel bandwidth or smoothing parameter.

`Exists any critical point`
: Emphasize if there is or not any critical.

Note

See more details in `change.points`.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas.

`plot.change.points` *Visualization of change.points objects*

Description

Useful for drawing the estimation and first derivative for each base pairs.

Usage

```
plot.change.points(x, base.pairs = NULL, der = NULL,
xlab = "x", ylab = "y", col = "black", ICcol = "grey",
main = "title", type = "l", ICtype = "l", ...)
```

Arguments

x	change.points object.
base.pairs	Character string about skew profile for A vs. T or C vs. G.
der	Number which determines inference process to be drawing into the plot. By default der is NULL. If it is 0, the plot represents the initial estimate. If der is 1, the first derivative is plotted.
xlab	Title for x axis.
ylab	Title for y axis.
col	A specification for the default plotting color.
ICcol	A specification for the default confidence intervals plotting color.
main	An overall title for the plot.
type	What type of plot should be drawn. Possible types are, p for points, l for lines, o for overplotted, etc. For more details par
ICtype	What type of plot should be drawn for confidence intervals. Possible types are, p for points, l for lines, o for overplotted, etc. For more details par
...	Other options.

Value

Simply produce a plot.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas

Examples

```
library(seq2R)
humanDNA <- read.genbank("NC_012920")
humanDNAbin <- change.binary(humanDNA)
hDNA<-change.points(humanDNAbin)
plot.change.points(hDNA,base.pairs="AT")
```

`critical` *Critical points (maximum and minimum).*

Description

Value of covariate `x` which maximizes and minimizes the first derivative of the model obtained with `change.points` function. Also, it is included their 95 % confidence intervals.

Usage

```
critical(x, base.pairs = NULL)
```

Arguments

`x`	`change.points` object.
`base.pairs`	Character string about for A vs. T or C vs G.

Details

In mitochondrial genomes, the trend of the skew profile curve changes abruptly at the replication origins, i. e. the concavity of skew profile should switch in this region (point of inflection). The first derivative of the skew profile curve will reach a maximum or minimum value at the location of replication origins (critical points).

Value

The returned list has two component (`$AT`, `$CG`). Both of them containing a matrix with values about their critical (maximum and minimum) points, lower and upper confidence intervals 95 %.

`AT`	Critical points for AT.
`CG`	Critical points for CG.

Author(s)

Nora M. Villanueva and Javier Roca-Pardiñas

Examples

```
# human mitochondrial DNA data
library(seq2R)
humanDNA <- read.genbank("NC_012920")
humanDNAbin <- change.binary(humanDNA)
hDNA<-change.points(humanDNAbin)
critical(hDNA)
```