eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

Máster Universitario en
Ingeniería **C**omputacional y **S**istemas **I**nteligentes

Konputazio Zientziak eta Adimen Artifiziala Saila –
Departamento de Ciencias de la Computación e Inteligencia Artificial

K I S A

I C S I

Tesis de Máster

DESARROLLO DE UN SISTEMA DE SIMULACIÓN DE DATOS DE METILACIÓN

MAITENA ITURRIBARRIA ASTORKIA

Tutor(a/es)

BORJA CALVO NAIARA G.BEDIAGA
Departamento de Ciencia de la Computación e Inteligencia Artificial
Facultad de Informática

informatika fakultatea
facultad de informática

KZAA /CCIA

Julio 2014

INDICE

TABLA DE ILUSTRACIONES

ABREVIATURAS

ADN : Ácido Desoxirribonucleico

ARN: Ácido Ribonucleico

ARNm: ARN mensajero

ADNc: ADN complementario

CpG : Dinucleótidos de Citosina-Fosfato-Guanina

A: Adenina

G: Guanina

T: Timina

C: Citosina

U: Uracilo

PCR: Polymerase Chain Reaction

GEO : Gene Expression Omnibus

MSP: Methylation Specific-Polymerase Chain Reaction

KDE: Kernel Density Estimation

RESUMEN

Hasta hace poco, enfermedades como el cáncer o el Alzheimer eran interpretadas solo como mutaciones genéticas, es decir, cambios en la secuencia genética. Sin embargo, son muchos los que últimamente se interesan por la epigenética y por la relación con las enfermedades. La epigenética va más allá que la genética, se basa en los cambios reversibles del ADN y de las proteínas que se unen en él. Esto hace que, sin necesidad de alterar su secuencia, un gen pueda ser expresado o por el contrario quede silenciado.

Uno de estos cambios epigenéticos es la metilación del ADN que consiste en una modificación química en el dinucleotido CpG (citosina-fosfato-guanina, es decir, donde una citosina es seguida de una guanina). Existen métodos experimentales para poder detectar la metilación, como por ejemplo, los métodos basados en la modificación del ADN con bisulfito y posterior análisis con arrays de ADN.

El objetivo de este proyecto es imitar, mediante la simulación computacional y el estudio de distintas bases de datos, el comportamiento del sistema biológico, a fin de generar datos similares a los reales. Esta simulación de los datos reales permitirá, entre otras cosas, generar escenarios controlados en los que evaluar los métodos de análisis. Adicionalmente, el proceso de diseño permitirá explorar el proceso biológico que da lugar a los datos.

SUMMARY

Until a few years ago, diseases such as cancer or Alzheimer's were interpreted only as genetic mutations, in other words, changes in the genetic sequence. However, lately interest has grown in epigenetics and its relationship with diseases. Epigenetics goes beyond genetics, as it is based on reversible changes in the DNA and the proteins joined to the DNA. This makes a gene to be expressed or, otherwise, silenced without altering the sequence.

One of these changes is called DNA methylation, which consists of a chemical modification in the CpG sites, (cytosine-phosphate-guanine sites, that is, where a cytosine is directly followed by a guanine). There are experimental methods to detect the methylation, for example, techniques based on bisulfite conversion and subsequent analysis using microarray technologies.

The goal of this project is to imitate the behaviour of a biological system using computer simulation and the study of various databases, in order to generate similar data to the real ones. This simulation of real data will allow, among other things, to generate controlled scenarios in which to evaluate the analysis methods. Additionally, the process of design will allow to explore the biological process that gives way to the data.

LABURPENA

Duela gutxi arte, minbizia eta Alzheimerra bezalako gaixotasunak mutazio genetiko gisa ezagutzen ziren, hau da, sekuentzia genetikoan aldaketak izatea bezala. Hala ere, asko dira azkenaldian epigenetika arloan eta harremana duen gaixotasunetan interesa dutenak. Epigenetika genetika baino haratago dago, DNAren eta bertan lotzen diren proteinen aldaketa itzulgarri oinarritzen baita. Honek gene bat egoera adierazgarri batetan edo bestalde egoera isil batetan egotea eragiten du, sekuentzia aldatu gabe.

Aldaketa epigenetiko bat DNAren metilazioa da, CpG (zitosina-fosfato-guanina, hau da, zitosina baten atzetik guanina bat datorrenean) posizioetan gertatzen den aldaketa genetiko batetan oinarritzen dena. Badaude metilazioari antzemateko zenbait metodo esperimental, esaterako, bisulfitoaren bidez eragindako DNAren aldaketatan eta microarrayetan oinarritutako metodoa.

Proiektu honen helburua sistema biologikoaren portaera imitatzea da, simulazio konputazionalaren eta datu base ezberdinen ikerketaren bidez. Helburua datu errealen antzeko datuak sortzea da. Datu errealen simulazio honek, analisi metodoak ebaluatzeko eszenatoki kontrolatuak sortzea ahalbidetuko du. Horrez gain, diseinu prozesuak datuak sortzen dituen prozesu biologikoa arakatzea ere ahalbidetuko du.

1.INTRODUCCIÓN

1.1.BIOLOGÍA, GENÉTICA Y EPIGENÉTICA

La Bioinformática trabaja en la investigación biológica y computacional para el desarrollo de herramientas que permitan analizar de una manera más eficiente la información biológica para, por ejemplo, percibir la influencia de las enfermedades con la ayuda de la información genética y las funciones y estructuras biológicas.[1]

En 1869, cuatro años después de que el monje austriaco Gregor J. Mendel definiera sus experimentos genéticos, Frederich Miescher, un famoso médico y biólogo suizo, descubrió por primera vez la nucleína. Miescher identificó en el núcleo de las células del pus un nuevo grupo de substancias celulares, ricas en fosforo, que mas tarde serían reconocidas como ADN o acido desoxirribonucleico. La relación entre la nucleína y la genética fue establecida pasado casi un siglo.

En 1952 James Watson y Francis Crick realizaron una serie de experimentos que condujeron a la determinación de la estructura del ADN. La estructura de la doble hélice está formada por dos hebras. Tiene forma de escalera enlazada y está compuesta por nucleótidos. Los nucleótido se componen de una molécula de azúcar, la desoxirribosa, un grupo fosfato y una base nitrogenada. En el ADN se distinguen cuatro nucleótidos dependiendo de la base nitrogenada que contiene: Adenina (A), Guanina (G), Timina (T) y Citosina (C).

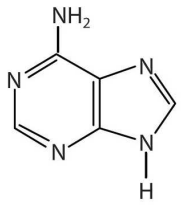
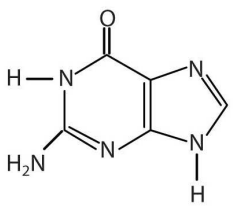
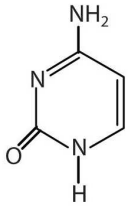
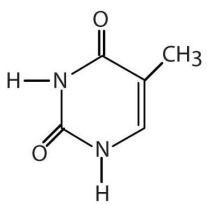
			
Adenina (A)	Guanina (G)	Citosina (C)	Timina (T)

Tabla 1-Bases nitrogenadas que forman los diferentes nucleótidos del ADN

El monosacárido se encuentra arropado por un grupo fosfato y una base en el centro del nucleótido. Las bases se complementan entre ellas, uniéndose por enlaces de tipo puente de hidrogeno; por un lado la Adenina se asocia con la Timina y por otro, la Guanina se enlaza con la Citosina.

Una de las funciones primordiales del ADN es codificar las proteínas, las cuales son moléculas poliméricas formadas por aminoácidos. En seres humanos existen 21 tipos de aminoácidos. Los aminoácidos se unen entre ellos formando cadenas para generar distintas proteínas. De este modo, cada proteína se diferencia según la secuencia de aminoácidos que contiene.

Se conocen entre 250.000 y 1.000.000 de proteínas diferentes. Gracias a la heterogeneidad de la estructura que estas pueden adoptar, las proteínas pueden desempeñar un número considerable de tareas de nuestro organismo.

Para poder sintetizar la cadena de aminoácidos, en primer lugar, la información contenida en la secuencia de ADN se transfiere a una molécula de ARN mensajero. A este proceso se le llama “transcripción”. Durante el proceso de transcripción, las secuencias de ADN son copiadas a ARN mediante una enzima llamada ARN polimerasa que sintetiza un ARN mensajero que mantiene la información de la secuencia del ADN. Luego, tras sufrir unos cambios, el ARNm es utilizado por los ribosomas para generar nuevas proteínas en un proceso llamado “traducción” [2].

Una tarea importante es el control de la expresión genética. Los factores de transcripción son, entre otros mecanismos, los encargados de este proceso. Son proteínas que se unen al gen y ayudan a iniciar los procesos de transcripción. Su función es reconocer las regiones promotoras, y unirse a ellas modulando la actividad de la ARN polimerasa. Otro tipo de mecanismo de regulación genética que se está investigando en los últimos años son los agrupados dentro de la epigenética.

Una de las modificaciones epigenéticas más comunes es la metilación del ADN. La metilación consiste en la transferencia de grupos metilos a algunas de las bases citosinas (C) del ADN situadas previa y contiguamente a una guanina (G). La metilación es fundamental en la regulación del silenciamiento de los genes, y en consecuencia está asociado a una amplia gama de los procesos biológicos y enfermedades. Así, por ejemplo, cumple un importante papel a la hora de mantener el silenciamiento génico en el desarrollo normal, la impronta genómica y la inactivación del cromosoma X. [3][4][5]

Un aspecto importante de la metilación del ADN es que estos cambios, pese a no alterar la secuencia, pueden ser transmitidos de generación en generación. Igualmente, pueden ser adquiridos con el paso de los años por efecto del entorno. Es como una pequeña memoria de los genes que, en cierta manera, puede llegar a guardar desde lo que comieron nuestros abuelos en la guerra hasta el estrés que sufrimos en el trabajo.

Se ha observado que diversas enfermedades humanas como el cáncer, enfermedades inflamatorias o las neuropsiquiátricas entre otras, muestran una alteración en sus patrones de metilación con respecto a los de otros individuos sanos. Además, la metilación de ADN se ha asociado a la evolución de algunas enfermedades o respuesta al tratamiento. Por este motivo, se ha propuesto como un marcador diagnóstico, pronostico o predictivo con un enorme potencial para su aplicación e la práctica clínica.

La metilación varía según el tipo de tejido en el que se mide, el tipo de enfermedad, la fase de la enfermedad, etc. Por ejemplo, diferentes estudios han demostrado que cada una de las células de cáncer de colon, mamario, próstata y pulmón tienen su propio sello epigenético.

1.2.TECNOLOGÍAS PARA LA DETERMINACIÓN DE LOS NIVELES DE METILACIÓN

Existen múltiples técnicas basadas en métodos enzimáticos y/o químicos para determinar y cuantificar el porcentaje de metilación del ADN y poder guardarlos computacionalmente. Una vez que los datos estén informáticamente guardados es más fácil poder analizar y obtener una estimación absoluta del nivel de metilación.

La elección de la tecnología a emplear viene condicionada por la región del genoma que se quiera abarcar, el numero de sitios CpG que se quiera testar, la sensibilidad y/o especificad requerida o el tipo de muestra a analizar(si está o no degradado, cantidad de ADN….).

Existen diversas metodologías para la determinación de los niveles de metilación. Estas se pueden clasificar de diferentes maneras. Se pueden clasificar en función del pre-tratamiento recibido (digestión enzimática, enriquecimiento por afinidad o tratamiento bisulfito), o también por las regiones del genoma que interrogan. De esta manera tenemos las técnicas locus específicas (HpaII-PCR, MethyLight, pirosecuenciación o EpiTYPER entre otros), las técnicas basadas en microarrays (DMH, MCAM, MethylScope, etc.) o aquellas basadas en secuenciación masiva (Methyl-seq, MSCC, etc.).

En el presente trabajo nos centraremos en aquellas técnicas cuyo tratamiento se basa en la modificación bisulfito y su posterior análisis se realiza mediante microarrays. A continuación veremos los pasos fundamentales para procesar una muestra, los cuales se resumen en la siguiente figura.

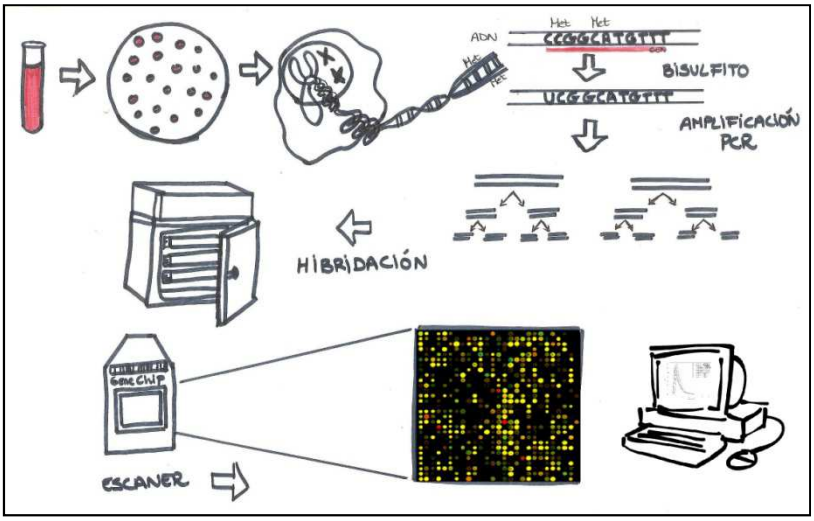
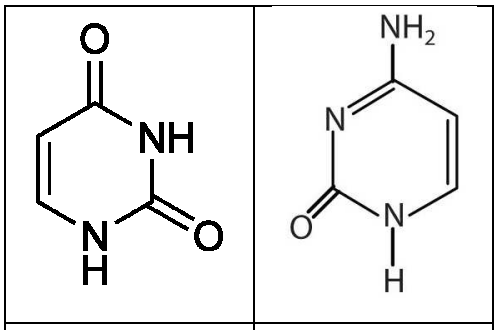


Ilustración 1-Pipeline de procesamiento de un estudio de metilación

1.2.1.PREPARACIÓN DE LAS MUESTRAS

El primer paso es la modificación bioquímica del ADN tratándolo con bisulfito de sodio. El bisulfito sódico actúa sobre el ADN, convirtiendo los residuos citosina no metilados en uracilos mediante deaminación.



Uracilo (U)	Citosina (C)

Como bien se ha explicado anteriormente, esto ocurre con las citosinas no metiladas, ya que las metiladas quedan protegidas de dicha transformación. Por lo tanto, el método del bisulfito de sodio produce cambios en la secuencia de ADN que dependen del estado de metilación de las citosinas. [6]

Tras la modificación, el ADN resultante se amplifica mediante el proceso PCR. En algunos casos se puede realizar una PCR específica que distingue las citosinas no metiladas de las metiladas; se consiguen resultados en un tiempo corto, pero esta técnica es más propensa a errores.

1.2.2. CONCEPTOS GENERALES DE LA HIBRIDACIÓN Y LOS MICROARRAYS

Las muestras pueden que ser analizadas computacionalmente, para ello se utilizan habitualmente arrays de ADN. Los arrays son cristales en los cuales hay una serie de pocillos. En el interior de estos pocillos se “pegan” secuencias de ADN conocidas.

Tras marcar con fluorocromos el producto de la PCR, este se pone en contacto con el array en un proceso llamado hibridación. Durante este proceso los fragmentos de ADN se unen a las secuencias específicas del array, quedando inmovilizadas.

Finalizado la hibridación los restos de PCR que no se han unido se lavan y el array se escanea para medir la luz emitida por los fluorocromos de las secuencias retenidas en cada pocillo. La intensidad de luz leída es proporcional a la cantidad de ADN unida.

Ilustración 2- Horno de Hibridación

1.3.ALGUNOS CONCEPTOS SOBRE PROBABILIDAD Y ESTADÍSTICA

Dentro del desarrollo experimental de este trabajo se han utilizado extensivamente las distribuciones de probabilidad. Estas distribuciones son funciones que asignan una probabilidad a cada suceso de la variable aleatoria. Esto es, definen para cada suceso la probabilidad de ser observado.

Una tarea particularmente importante en el contexto de este proyecto es el muestreo de las distribuciones de probabilidad. Es decir, conocida la distribución de una variable, generar datos que sigan dicha distribución.

Las distribuciones que se han utilizado en este trabajo son la distribución binomial, la distribución beta, la distribución de Dirichlet y la distribución normal.

1.3.1.DISTRUBUCION BETA

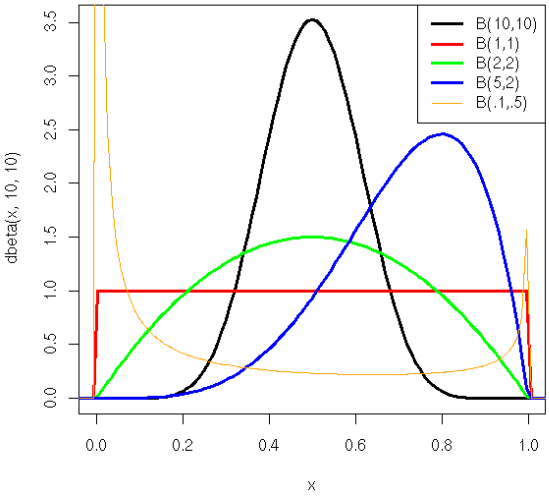
La distribución Beta es una distribución de probabilidad continua que hace uso de dos parámetros, α y β. La función de densidad, que se muestra a continuación, está definida en el intervalo x∈[0,1].

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, x^{\alpha-1}(1-x)^{\beta-1}$$

La ecuación está definida en términos de la función gamma Γ, que es la generalización del factorial y cuya definición es:

$$\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx$$

A continuación se muestran varios ejemplos de distribución betas (Figura 3).



α, β >1: Forma de campana.

α=β=1: Uniforme.

α=1,β >1 ó α >1,β=1: Exponencial, valor finito en 0 y 1.

α<1,β >1 ó α >1,β<1: Exponencial, valor infinito en 0 ó 1.

Ilustración 3- Representación gráfica de varias Distribuciones Beta

Parte del trabajo se basa en la búsqueda de los parámetros a partir de una muestra. Una forma de hacerlo es usando la media y la varianza, cuyas ecuaciones son:

$$E[X] = \frac{\alpha}{\alpha + \beta} \quad , \qquad V[X] = \frac{\alpha\,\beta}{(\alpha + \beta + 1)(\alpha + \beta)^2}$$

1.3.2.DISTRIBUCION DIRICHLET

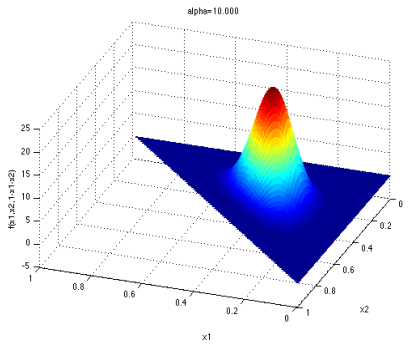
La distribución Dirichlet es una distribución de probabilidad multidimensional continua definida sobre vectores de números reales positivos tales que su suma es uno. Se puede ver como la generalización de la Beta.

Si X~Dirichlet($\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k$) entonces:

$$f(x; \alpha_1, \alpha_2, \dots, \alpha_k) = \frac{\Gamma(\alpha_1 + \alpha_2 + \cdots + \alpha_k)}{\Gamma(\alpha_1)\Gamma(\alpha_2) \dots \Gamma(\alpha_k)}\, x_1^{\alpha_1 - 1} \dots x_k^{\alpha_k - 1}$$

$$\sum_{i=1}^{k} x_i = 1 \qquad \forall_i\, x_i \geq 0$$

La siguiente figura muestra un ejemplo de distribución Dirichlet bidimensional:



Como puede apreciarse, la función solo está definida en el triangulo formado por los $x_1\ x_2$ tales que $x_1 + x_2 = 1$.

Ilustración 4- Distribución Dirichlet Representación gráfica 3D

1.3.3.DISTRIBUCION BINOMIAL

La distribución binomial es una distribución de probabilidad discreta que modela la cantidad de éxitos de *n* ensayos de Bernoulli independientes entre sí.

Dentro de un experimento de Bernoulli solo puede haber dos resultados, éxito, con una probabilidad *p*, o fracaso, con una probabilidad 1-*p*. En el caso de la binomial, este experimento se repite *n* veces para finalmente calcular la probabilidad de un número determinado de éxitos. Por tanto los parámetros de la distribución binomial son *n* y *p*.

Para explicar mejor este concepto, pongamos como ejemplo el lanzamiento de una moneda varias veces, considerando éxito la obtención de cara. Supongamos que se lanza una moneda diez veces. La probabilidad de tener cuatro caras, (y por tanto seis cruces), será $\binom{10}{4} p^4(1-p)^6$ donde p es la probabilidad de tener cara. En general, para una binomial *n p*, la probabilidad de que la variable tome el valor *x* es:

$$p(x;n,p) = \binom{n}{k} p^x (1-p)^{n-x}$$

1.3.4.DISTRIBUCIÓN NORMAL

La distribución de probabilidad más conocida es la distribución normal también conocida por campana de Gauss, ya que al representar la función de probabilidad tiene forma de campana.

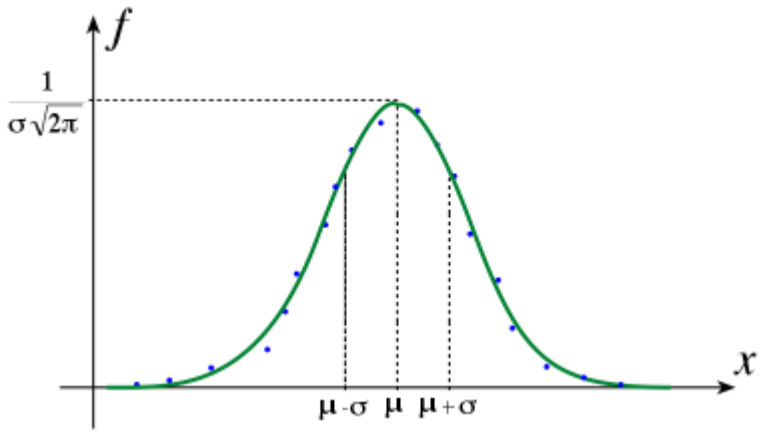


Ilustración 5- Distribución normal

La distribución normal se define en función de dos parámetros, la media μ y la desviación típica σ y se designa por N(μ, σ). La media, mediana y moda de la distribución normal es igual y se localiza en el centro de la distribución. Al ser una distribución simétrica respecto a la media, deja a un área igual en un lado que en el otro.

La ecuación de la distribución es el siguiente:

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

1.3.5.ESTIMACIÓN DE DENSIDAD BASADA EN KERNELS

Una de las formas más habituales de estimar la función de densidad a partir de una muestra son las densidades basadas en kernels (en inglés, KDE, Kernel Density Estimation).

Las KDE son estimaciones no paramétricas de la densidad que utiliza funciones kernel. El objetivo es construir una función de densidad teniendo en cuenta los valores muestrales de la siguiente manera.

$$f(x) = (n\,h)^{-1} \sum_{i=1}^{n} K\left((x - x_i)h^{-1}\right)$$

donde:

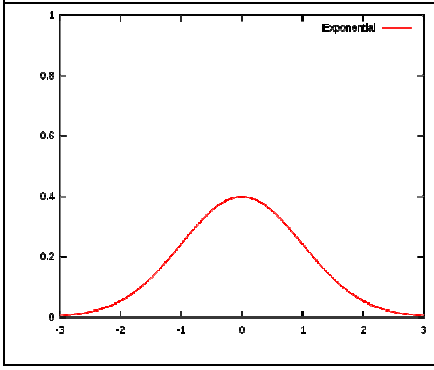
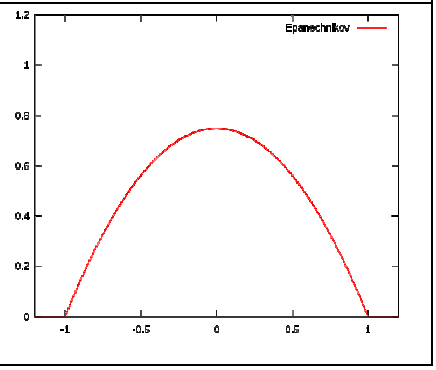
$(x_1, x_2, ..., x_n)$ = una muestra aleatoria de la distribución a estimar

n = tamaño muestral

h = ancho de banda o parámetro de suavizado

K = función Kernel

El valor h, conocido como ancho de banda, es el valor que determina la zona de influencia de cada punto de la muestra. Como función kernel (K) se pueden utilizar distintas funciones, tales como la función de densidad Gaussiana o la función de Epanechnikov.

GAUSS	EPANECHNIKOV
$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$	$K(u) = \begin{cases} \frac{3}{4}(1-u^2) & \{\lvert u\rvert \le 1\} \\ 0 & en\ otro\ caso \end{cases}$
	

A continuación se muestra gráficamente un ejemplo de este tipo de estimación. Las cruces situadas en y=0 representan los puntos de la muestra mientras que las curvas con trazo discontinuo representan las funciones kernel situadas en dichos puntos; la línea de trazo continuo es la estimación de la densidad.

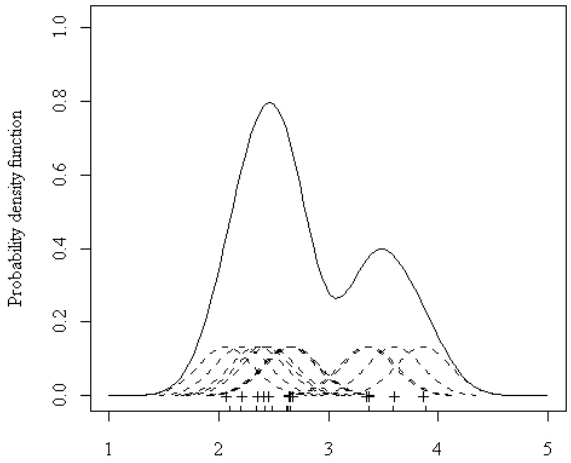


Ilustración 6-KDE Kernel Density Estimations

1.4.MOTIVACIÓN Y OBJETIVOS DEL PROYECTO

Durante la última década la tecnología de generación de datos en biología ha evolucionado a pasos agigantados; la necesidad de analizar los grandes volúmenes de datos ha dado lugar a la creación de una nueva disciplina científica, la bioinformática.

Esta revolución ha permitido generar inmensas cantidades de datos, permitiendo que los investigadores puedan estudiar la biología humana desde otra perspectiva y entender las patologías moleculares. Para ello, los datos son analizados mediante el uso de avanzadas técnicas de minería de datos, estadísticas, etc. para extraer conocimiento útil de ellos.

Mientras tanto, evaluar los métodos de análisis de datos supone un gran problema, puesto que es imposible conocer la información que se esconde en los datos reales. Como solución se plantea la simulación, esto es, emular computacionalmente el comportamiento biológico. Por ello, en este trabajo el objetivo es explorar la generación de datos de metilación.

1.4.1.OBJETIVO GENERAL

El objetivo principal de este estudio es imitar, mediante la simulación computacional y el estudio de distintas bases de datos, las mediciones obtenidas de los datos reales.

1.4.2.OBJETIVOS ESPECIFICOS

- Entender el proceso biológico y pensar como simularlo computacionalmente.
- Analizar con cautela las diferentes bases de datos de GEO con grandes cantidades de genes e interpretar los datos reales.
- Diseñar un script que simule el comportamiento de la metilación del ADN de un site concreto para diferentes individuos.
- Evaluar el valor de cada parámetro y ver su influencia para ajustarlos hasta que los datos se parezcan a los datos reales.
- Desarrollar un script para el muestreo de un número arbitrario de posiciones CpG.
- Valorar los resultados y plantear mejoras si se puede.
- Crear un paquete de R para que distintos usuarios puedan utilizarlo de una manera fácil y eficiente.
- Explicar todo lo realizado en el trabajo.

1.5.ESQUEMA DEL DOCUMENTO

El esfuerzo de este trabajo está orientado a la imitación del comportamiento del sistema biológico. La presentación del trabajo se divide en 3 secciones principales, la primera introduce al lector a conocer los aspectos generales acerca del tema de este trabajo. Se expone una breve introducción a la biología y a las diferentes tecnologías necesarias para poder estudiar el comportamiento de la metilación. En este primer apartado se plantean los objetivos específicos y generales.

La segunda sección, la más importante, se dedica principalmente al trabajo realizado. Este apartado se distribuye en 5 subsecciones. En primer lugar, en la sección 2.1, explica cómo se obtienen los datos, esto es, el análisis exploratorio de los datos públicos, en el cual se hace uso de diversas bases de datos. Es una de las partes más importantes del trabajo, ya que es el comienzo y la que da pie a poder realizar el resto del trabajo.

En segundo lugar, en la sección 2.2, se describe globalmente el proceso biológico desde la muestra hasta los datos. Es necesario entender cada uno de los procesos y técnicas que se utilizan en el estudio de la metilación, para luego poder simularlo computacionalmente.

Después, en la sección 2.3, se expone detalladamente todo el desarrollo del sistema. El objetivo es imitar el comportamiento del sistema biológico. Dentro del desarrollo se han utilizado extensivamente la estadística y la probabilidad.

Luego, en la sección 2.4, se explica el análisis de todos y cada uno de los parámetros del sistema, comparándolos con los datos reales. Es interesante ver los resultados logrados y contrastarlos con los resultados reales.

Finalmente, en la sección 2.5, se explica brevemente la implementación del sistema y como puede usarlo el usuario.

Por último, en la sección 3, se presentan las conclusiones obtenidas. Una argumentación clara y concisa de lo que consideramos después de analizar y obtener una serie de respuestas.

2.DESARROLLO

2.1.ANÁLISIS DE LOS DATOS REALES

Existen diversas herramientas software para realizar experimentos y análisis de datos. Uno de los softwares más conocidos en el campo de la estadística y la bioinformática es R, que hace uso de su propio lenguaje de programación. Una de las mayores ventajas es que dispone de muchísimos paquetes; además es un software de código abierto lo que hace que sea muy accesible.

Hay dos bases de datos principales que contienen datos de metilación: GEO (Gene Expression Omnibus)[1] y ArrayExpress[2]. Para este trabajo hemos utilizado bases de datos reales pertenecientes a GEO.

El análisis de perfil de metilación se ha realizado sobre datos generados con arrays Infinium HumanMethylation27 bedchip de la compañía Illumina. Son arrays que examinan el estado de metilación de más de 27000 posiciones CpG o sites. Se han analizado cuatro bases de datos de cuatro tejidos diferentes:

- GSE49904: sangre (27.578 sites y 71 individuos)
- GSE49905: cerebro (27.578 sites y 78 individuos)
- GSE49907: riñón (27.578 sites y 83 individuos)
- GSE49908: músculo (27.578 sites y 51 individuos)

Cada una de las bases de datos contiene los valores beta de cada individuo en cada posición CpG. Este valor representa el ratio de moléculas en las que el site está metilado y, por tanto, toma valores entre 0 y 1.

El valor beta se estima de la relación entre la intensidad metilada y la intensidad global, esto es, la suma de la intensidad metilada y la no metilada; por lo general, más del 95% de los valores de la intensidad total son mayores que 1000.

Como primer análisis exploratorio de los datos, hemos estimado la distribución de los niveles de metilación globales usando estimadores basados en kernels. Se pueden ver los resultados en la Figura 7.

[1] http://www.ncbi.nlm.nih.gov/geo/
[2] http://www.ebi.ac.uk/arrayexpress/

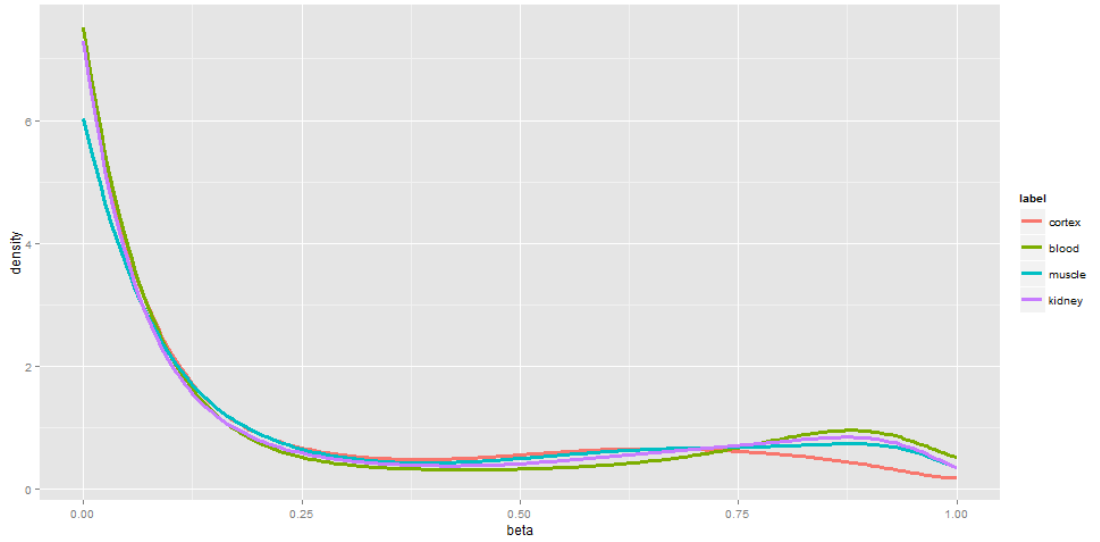


Ilustración 7- Distribución de los niveles de metilación

Lo primero que se puede apreciar es que la mayoría de los valores se concentran en torno al cero. No obstante hay unos cuantos que se acercan hacia uno. Dicho de otra manera se puede observar que hay posiciones que están metiladas, pero la gran mayoría no lo están. No obstante, también se pueden apreciar valores beta que tienen un valor intermedio.

Aunque en la imagen no se puede verse, es importante destacar que hay posiciones con valor cero, pero por el contrario no existen posiciones con valor uno.

Por la naturaleza de los valores hay una clara relación entre la metilación media y la varianza. Por ello, como segundo análisis hemos analizado gráficamente esta relación, que se muestra en las Figuras 8 a 11.

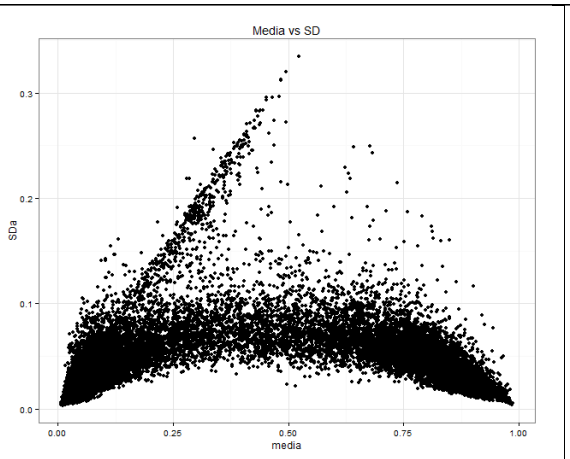


Ilustración 8-Relación entre media y varianza de la base de datos GSE49904 (sangre)

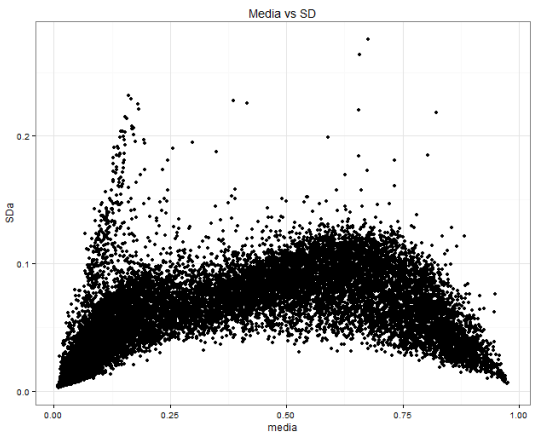


Ilustración 9- Relación entre media y varianza de la base de datos GSE49905 (cerebro)

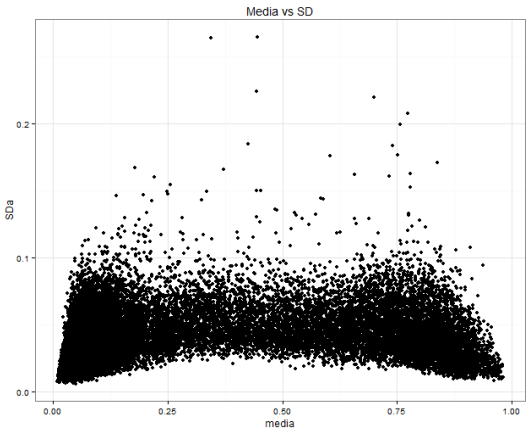


Ilustración 10- Relación entre media y varianza de la base de datos GSE49908 (músculo)

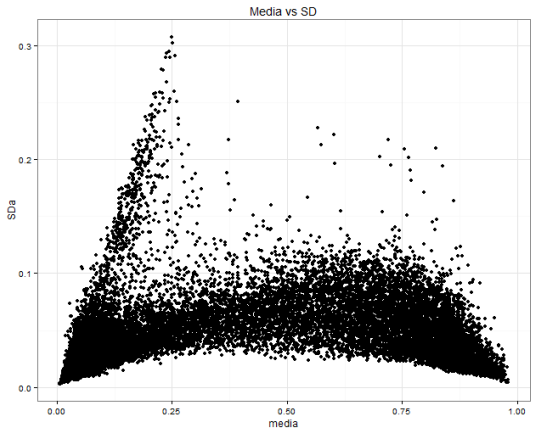


Ilustración 11- Relación entre media y varianza de la base de datos GSE49907 (riñón)

Estos últimos gráficos nos muestran que la mayoría de los valores de la variable SD o desviación estándar oscilan entre 0 y 0.1. Sin embargo, hay algunos datos que tienen una desviación entre 0.1 y 0.3.

Los gráficos muestran una forma curvada muy parecida a la del arco iris. Esto es debido a que las posiciones cuya media está más cerca del valor cero o uno tienden a tener, como es lógico, menos varianza.

Para ver mejor los valores de la desviación, se han representado gráficamente en la Figura 12 la distribución de la desviación estándar para las cuatro bases de datos.

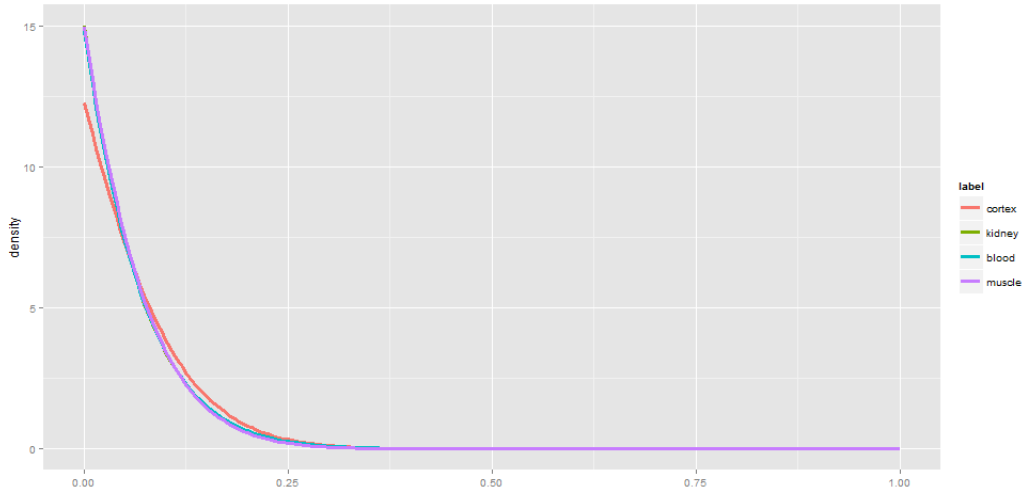


Ilustración 12- Valores de la desviación estándar en las cuatro bases de datos

Las observaciones empíricas sugieren que la variabilidad es distinta en las posiciones metiladas y las no metiladas. Por ello, se ha decidido representar y comparar los datos de una de las bases de datos, teniendo en cuenta los valores de media metilada y demetilada. Esto es, se han ilustrado aquellos valores de la desviación estándar que tuviesen un valor beta medio mayor que 0.8 y por otro lado, una media menor que 0.2.

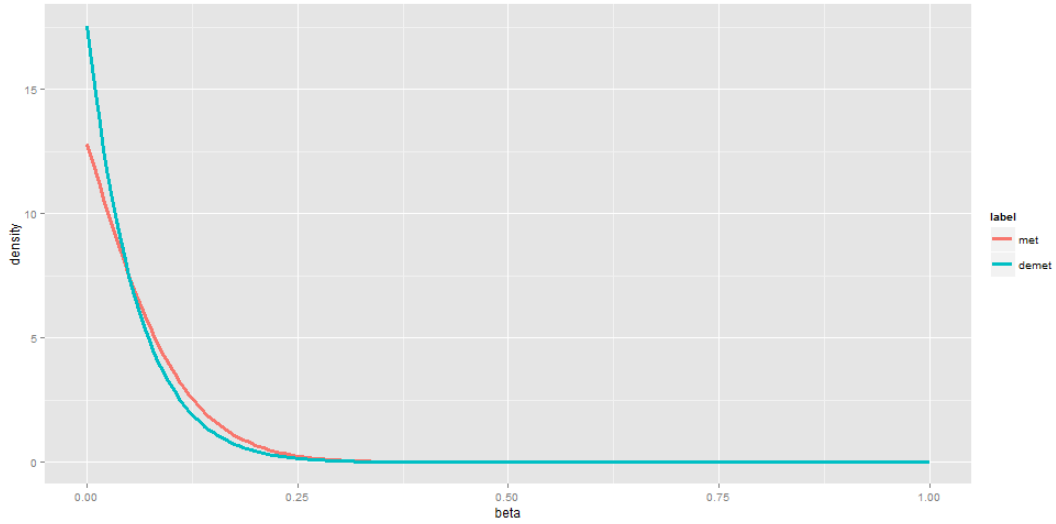


Ilustración 13- Desviación estándar de media mayor que 0.8 y menor que 0.2 de la base de datos cortex

Mediante este gráfico, (Figura 13), podemos observar que las posiciones demetiladas tienen una desviación menor aún, que las posiciones metiladas. La figura muestra los resultados para la base de datos de cortex; los resultados obtenidos en el resto son similares.

Para concluir, es importante entender la distribución de los valores beta de cada posición en distintos individuos. Por este motivo, se han representado una serie de gráficos para analizar mejor el comportamiento de cada site o posición.

A continuación se muestran algunos ejemplos significativos donde cada ilustración pertenece a una base de datos diferente y cada punto representa a un individuo en ese site concreto. Se pueden ver los gráficos en las Figuras 14 a 17.

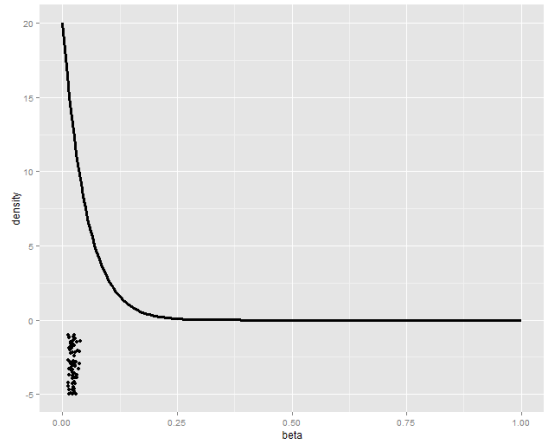


Ilustración 14- Site número 8 de la base de datos GSE49904 (sangre)

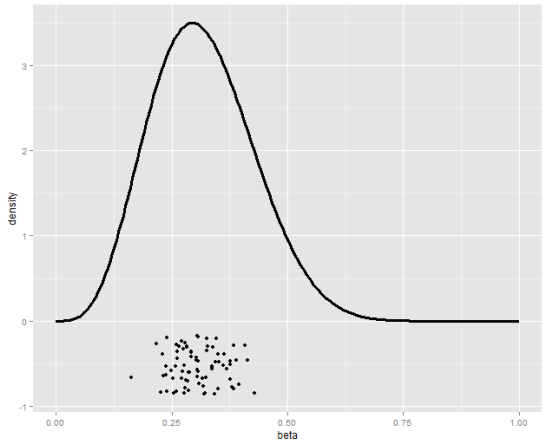


Ilustración 15- Site número 1534 de la base de datos GSE49905 (cerebro)

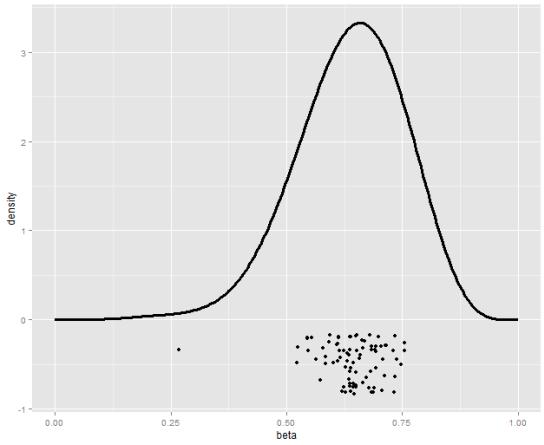


Ilustración 16- Site número 85 de la base de datos GSE49907 (riñón)

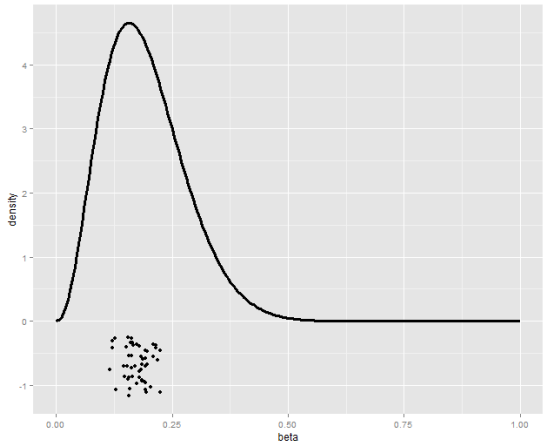


Ilustración 17- Site número 45de la base de datos GSE49908 (músculo)

En este punto de la investigación resulta relevante meditar sobre estos datos. Y es que nos preguntamos, ¿tiene algún sentido biológico que haya datos en donde, sistemáticamente, los individuos en un site concreto tenga, por ejemplo, un 25% de moléculas metiladas? ¿Cuál es la causa de esto?

En muestras tales como las PBMCs (peripheral blood mononuclear cells), se sabe que hay distintos tipos celulares, tales como linfocitos T, linfocitos B, células NK, monocitos, dendritas, etc. con diferentes proporciones.

Nuestra hipótesis es la siguiente ¿podrían ser que las metilaciones intermedias sean debidas a las mezclas de células metiladas y no metiladas? Es decir, un site en un determinado tipo celular solo puede estar metilado ($ratio \cong 1$) o demetilado ($ratio \cong 0$), y el resto de situaciones son debidas a diferentes estados en las células que forman la mezcla.

2.2.DESCRIPCIÓN GLOBAL DEL PROCESO BIOLÓGICO DESDE LA MUESTRA HASTA LOS DATOS

Exploremos un poco la idea de que en una muestra haya más de un tipo celular, cada uno representado en una proporción diferente. Esto es, la cantidad de células de cada tipo en esa muestra puede variar.

Así pues, si tenemos una muestra compuesta por dos células, estando una metilada y presente en una proporción del 25% y la otra demetilada con una proporción del 75% obtendremos como resultado una metilación observada de aproximadamente 0.25. A continuación se muestra gráficamente un ejemplo un poco más complejo (Figura 18).

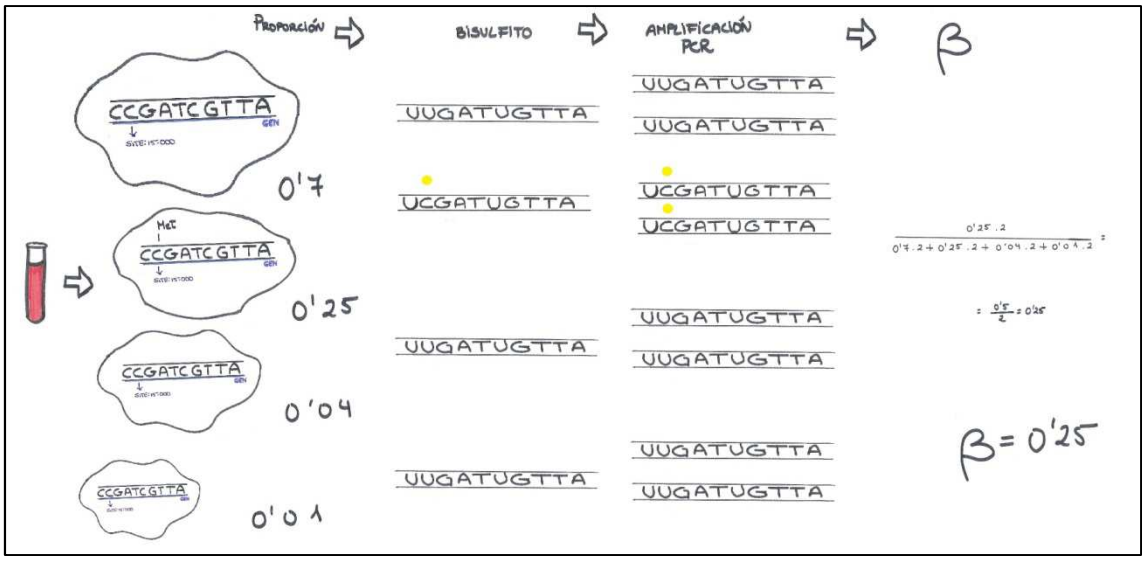


Ilustración 18- Efecto de la mezcla de tipos celulares

Otro fenómeno que es importante entender es la imprintación. Cada individuo hereda una copia del genoma de cada progenitor. En la mayoría de los casos la metilación de ambos coincide para todas las posiciones CpG, pero hay excepciones en que el individuo tiene una copia metilada y otra demetilada. Dicho de otra manera, el 50% de las moléculas estarán metiladas y el otro 50% no lo estarán. A este fenómeno se le llama imprintación y ocurre en un porcentaje bajo de posiciones.[7]

Resumiendo, en una muestra tendremos una mezcla de células en diferentes proporciones, cada una de las cuales estará metilada o demetilada. Además para cada gen tendremos dos copias que, en general, estarán en el mismo estado, salvo en un pequeño porcentaje de las posiciones.

Para preparar las muestras para ser analizadas en primer lugar, se deben procesar utilizando bisulfito sódico. Su función es convertir los residuos de citosinas no metiladas en uracilos. Esto solo ocurre con las citosinas no metiladas, ya que las metiladas quedan protegidas de la trasformación.

En segundo lugar, es necesario amplificar la muestra, para ello, el método más usado es la PCR.

El siguiente paso es la determinación del nivel de metilación con microarrays. El resultado de la PCR se marca con fluorescencia y se hibrida el array. Por último, los valores se leen por medio de un escáner.

Durante todo el proceso de generación de datos se generan errores. Algunos de estos errores son aleatorios, como por ejemplo los debidos a la lectura de la intensidad de un pocillo, mientras que otros pueden ser sistemáticos, como los debidos a una baja eficiencia en la conversión por bisulfito. Todos estos errores hay que tenerlos en cuenta a la hora de generar datos de forma sintética.

2.3.DESARROLLO DEL SISTEMA

Como primer paso, se han realizado todos los procesos teniendo en cuenta que se simula un único site o posición CpG.

En una muestra se ha considerado que se encuentran distintos tipos de células, cada uno en una proporción diferente. Así pues, se creará un tipo celular o clon al que llamaremos clon maestro, y posteriormente se crearán los clones restantes como copias ligeramente modificadas. El clon maestro es el clon con mayor peso en la muestra.

Un clon estará o bien metilado o bien demetilado. Para poder simular esto se han utilizado distribuciones Beta con forma exponencial. La distribución Beta es una buena aproximación para modelar el nivel de la metilación, ya que esta acotada de forma natural en el intervalo [0,1].

Con el fin de generar estos valores beta son necesarios los parámetros α y β. En el caso de que el clon esté metilado, entonces fijaremos β=1, y por el contrario, si el clon está demetilado, α=1. Por supuesto, una forma de fijar el otro parámetro es en base a la varianza, utilizando la ecuación:

$$\vartheta^2 = \frac{\alpha\,\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

La determinación de si el site está metilado o no se hace al azar, con una cierta probabilidad de estar metilado.

Con el propósito de lograr más clones, el proceso será similar exceptuando que al principio se cambiará el estado de metilación respecto al clon maestro, es decir, se intercambiarán α y β con una cierta probabilidad.

Por otro lado, cada clon, tendrá un peso asociado. En relación con esto, hay que generar distintas proporciones para cada tipo de célula en la muestra. Las proporciones serán aleatorias pero estarán controladas para que simulen a una muestra real.

Llegados a este punto, ya tenemos una muestra compuesta por distintas células, cada una con su proporción correspondiente. Para muestrear estas proporciones se utilizará la distribución de Dirichlet, donde los parámetros α_i representarán la relación entre las proporciones de cada clon.

$$(w_1, w_2, \dots, w_k) \sim Dirichlet(\alpha_1, \alpha_2, \dots, \alpha_k)$$

La distribución de Dirichlet se puede muestrear fácilmente mediante la utilización de la distribución gamma, ya que:

$$w_i \propto Gamma(\alpha_i, 1)$$

Como punto de partida para generar los parámetros tenemos los pesos canónicos $(r_1 \dots r_n)$ y un tamaño muestral equivalente α_t. Con ello, los parámetros se pueden calcular como:

$$\alpha_i = \frac{r_i * \alpha_t}{\sum_{j=1}^{k} r_j}$$

A la hora de muestrear el modelo, primero se conseguirá un valor beta por cada clon e individuo, muestreando la distribución Beta definida por sus parámetros α y β. Luego, se multiplicará cada valor Beta con su correspondiente peso muestreado de la distribución de Dirichlet. Finalmente, se sumarán para obtener el valor beta correspondiente a cada individuo permitiendo este proceso lograr los valores para todos los individuos.

En este punto, tendremos los valores beta correspondientes a un site en concreto y un individuo concreto. Los valores beta de todas las posiciones CpG de un individuo generan lo que llamamos un modelo de individuo.

El siguiente paso consiste en generar, partiendo de los valores beta, los valores correspondientes al número de moléculas metiladas (M) y demetiladas (U). Para ello, es necesario fijar el número total de moléculas (N). Para lograr esto, nos basamos en el hecho de que en la mayoría de los casos, la intensidad total es mayor que 1000. De todos modos, en la experimentación hemos usado un valor más grande (10000) para reducir la variabilidad debida a este paso. La determinación de la N se hace, por tanto, muestreando una distribución normal con μ=10000 y ϑ=100.

Una vez determinada la N, se usará la distribución binomial, de parámetros β,N. Para cada posición CpG se logrará la cantidad de moléculas metiladas (M) y las no metiladas (U), N=M+U. Los datos metilados seguirán la distribución binomial con la probabilidad de éxito igual al nivel de metilación muestreado, esto es, el modelo de individuo creado anteriormente y las muestras N.

Durante todo el proceso químico y tecnológico es probable que sucedan diversos errores. Algunos de estos errores, son aleatorios, como por ejemplo los debidos a la lectura de la intensidad de un pocillo, mientras que otros pueden ser sistemáticos, como los debidos a una baja eficiencia en la conversión por bisulfito.

Para atender a la tasa de errores que se puedan generar, es necesario aplicar un leve ruido en los datos. Por un lado, como ruido de mala lectura del escáner, al resultado de la intensidad, se le ha aplicado la distribución normal con:

$$\mu = intensidad\ metilada \qquad \vartheta = intensidad\ metilada/S_k$$

El proceso se repetirá para los valores de intensidad no metilada. Esto hará, que la variabilidad de los datos aumente un poco. Por otro lado, debido al background que puede haber en los arrays, al resultado de la intensidad se le ha añadido ruido blanco, es decir, un valor obtenido de una distribución normal con los siguientes parametros:

$$\mu = 0 \qquad \vartheta = desviación\_background$$

En este caso, la distribución Gaussiana hará mover un poco la línea hacia uno de los lados. Por último se le añadirá un porcentaje de error de eficiencia por la conversión de bisulfito.

Una vez simuladas las intensidades I_M e I_U, el valor Beta final se determina como:

$$beta = \frac{I_M}{I_M + I_U + offset} \qquad offset = 100$$

El valor de desplazamiento, "offset", teniendo en cuenta que los valores de intensidad son mayores que 10000, un valor como 100 es relativamente pequeño y tiene un efecto despreciable sobre el valor obtenido.

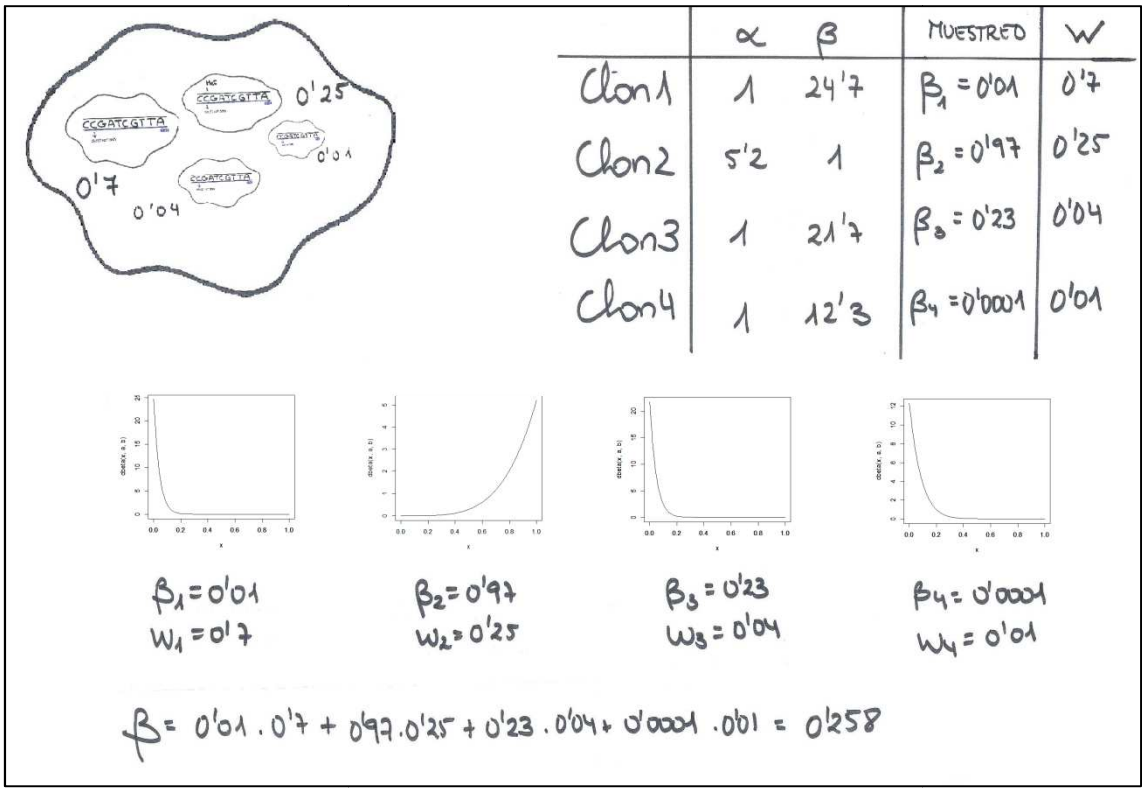


Ilustración 19-Esquema del sistema

2.4.ANÁLISIS DE LOS PARÁMETROS DEL SISTEMA Y COMPARACIÓN CON DATOS REALES.

Los parámetros del sistema descritos en el apartado anterior son:

- Probabilidad de metilación: Probabilidad para la determinación de si el site debe estar metilado o no.
- Rango Alpha metilado: Rango de valores para determinar el valor de α de la distribución Beta cuando $\beta=1$.
- Rango Beta demetilado: Rango de valores para determinar el valor de β de la distribución Beta cuando $\alpha=1$.
- Probabilidad de cambio: Probabilidad para la determinación de si el nuevo clon debe de ser modificado respecto al clon maestro.
- Ruido Escáner: Se aplicará un ruido Gaussiano para simular los errores de escáner que pueden haber. La desviación de esta distribución será el resultado de dividir las intensidades metiladas y demetiladas respecto a este parámetro.
- Ruido Background: Se aplicará un ruido Gaussiano para simular los errores del background. Este parámetro será el que determine la desviación de dicha distribución ya que, la media se fija a 0.

A continuación podemos ver los resultados de nuestro trabajo. Para determinar el valor ideal de las variables se han realizado una serie de pruebas. Todas las pruebas en un principio se han hecho con 70 individuos, como en la mayoría de las bases de datos reales, 4 clones y 100 posiciones CpG, para reducir el coste computacional.

Para empezar, ha sido necesario realizar una prueba con los valores que intuíamos, a nivel biológico, que eran los más razonables. En esta primera ejecución se ha estimado la distribución de los niveles de metilación usando estimadores basados en kernels, al igual que con los datos reales.

En esta primera aproximación basada en conocimiento experto, hemos utilizado los siguientes valores de parámetros, probabilidad de metilación=0.25, rango Alpha metilado=[2,5], rango Beta demetilado=[2,5], probabilidad de cambio=0.2, ruido escáner=10000, ruido background=0.001:

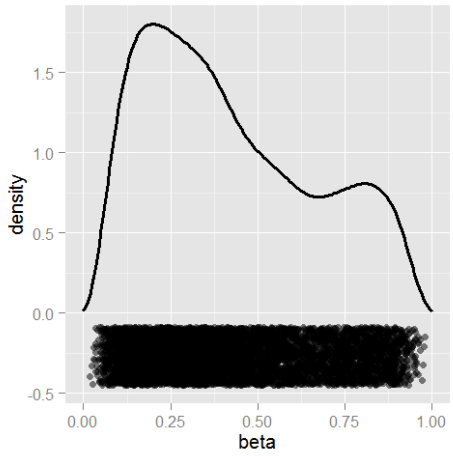


Ilustración 20-Primera representación gráfica

Como se puede apreciar en la Figura 20 los resultados en un comienzo no son muy buenos comparando con los datos reales. Con el propósito de mejorar los resultados, cada variable ha sido analizada y modificada repitiendo el proceso de ejecución una y otra vez hasta lograr un conjunto de datos parecido a los datos reales.

Después de realizar muchas pruebas se han tomado como referencia los siguientes valores de los parámetros, ya que los resultados logrados se asemejan bastante a los datos reales: probabilidad de metilación=0.2, rango Alpha metilado=[3,10], rango Beta demetilado=[3,35], probabilidad de cambio=0.05, ruido escáner=10000, ruido background=0.001.

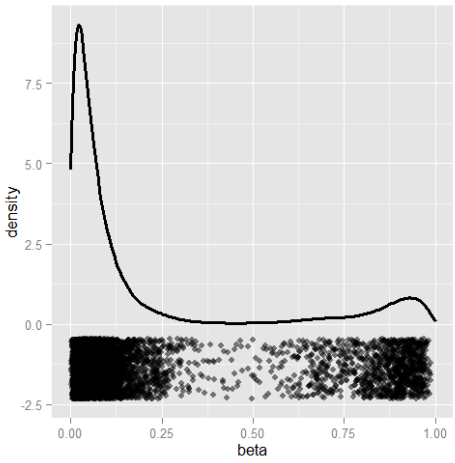


Ilustración 21-Resultados próximos a los datos reales

A fin de entender el efecto que cada parámetro tiene en el resultado, vamos a observar las diferencias entre distintos valores para los parámetros, modificándolos individualmente (Figura 22-33).

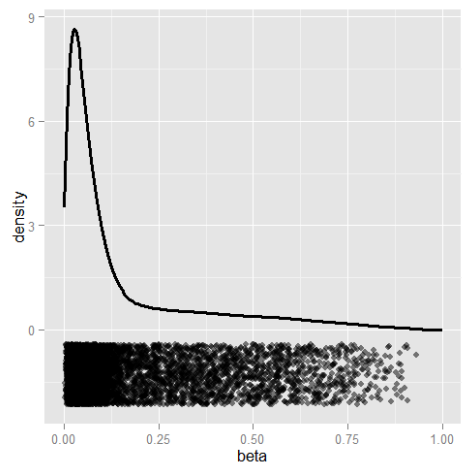
<table>
<tr><td>Probabilidad de Metilación</td><td> 

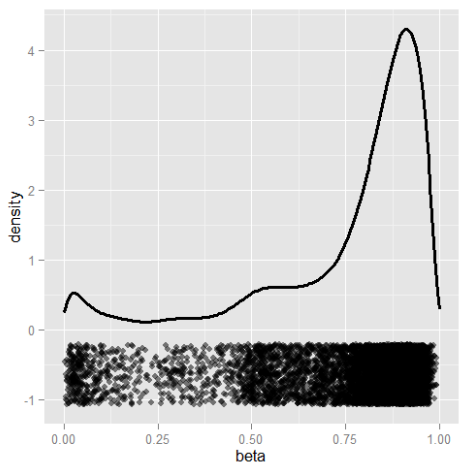
Ilustración 22- Probabilidad de Metilación Baja (0.001)</td><td> 

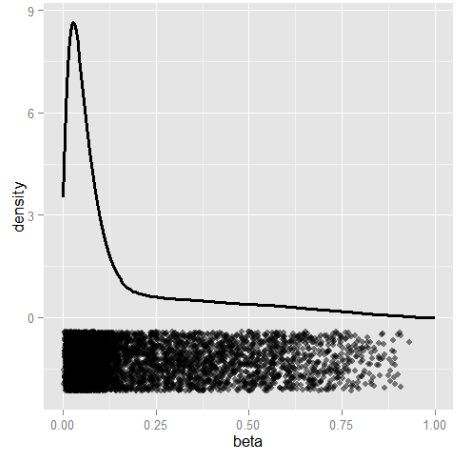
Ilustración 23- Probabilidad de Metilación Alta (0.9)</td></tr>
<tr><td>Rango α</td><td> 

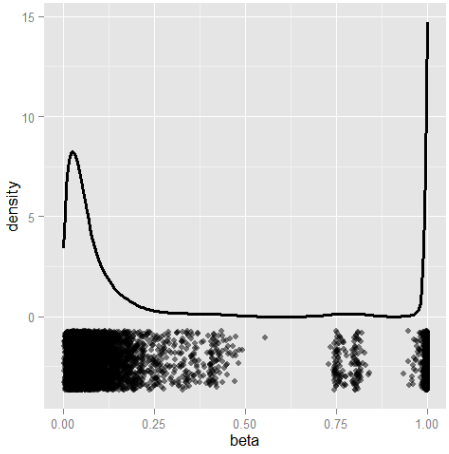
Ilustración 24- Rango Bajo ([0.3,2])</td><td> 

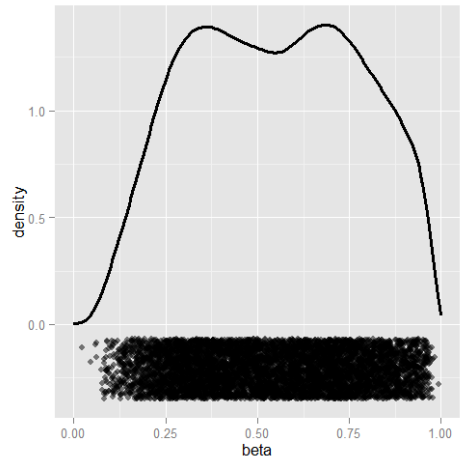
Ilustración 25- Rango Alto ([3,1000])</td></tr>
<tr><td>Rango β</td><td> 

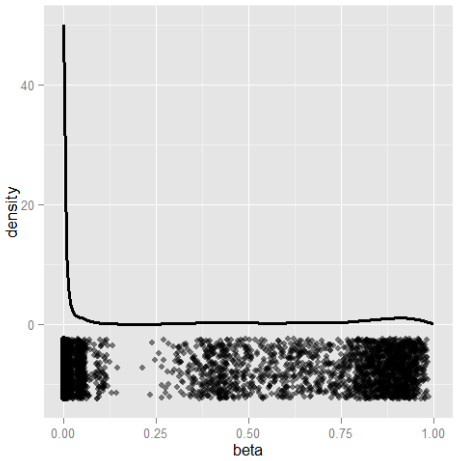
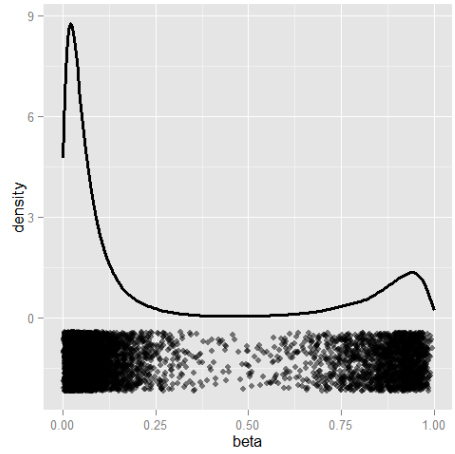
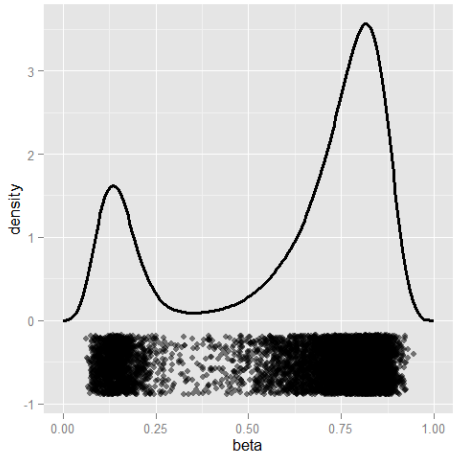
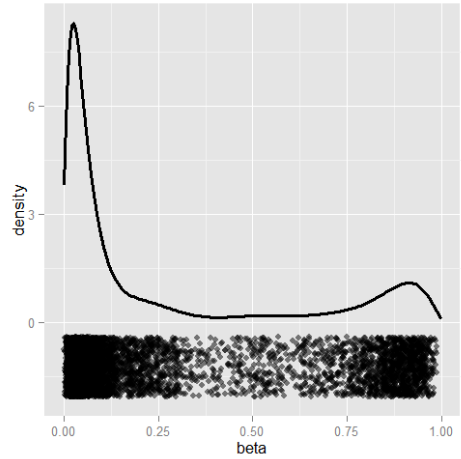
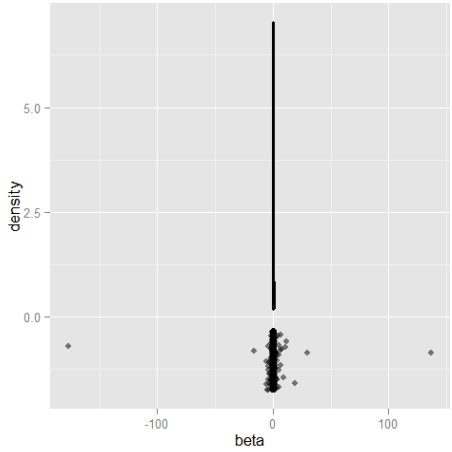
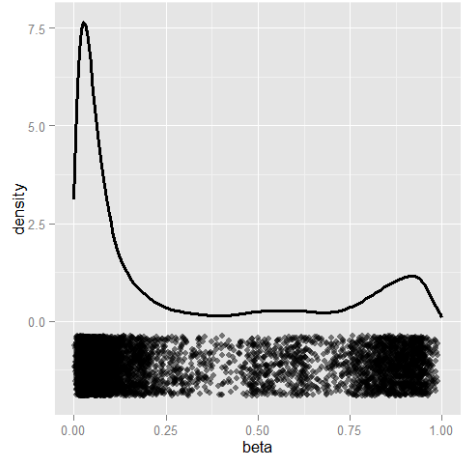
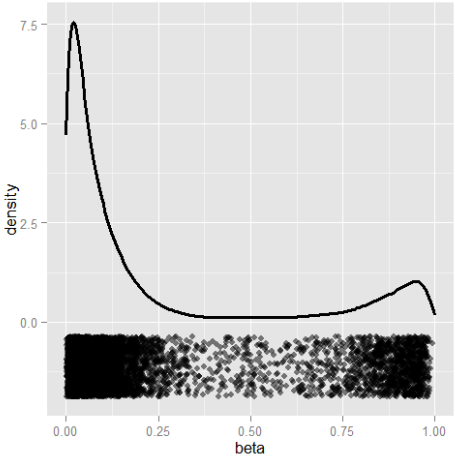
Ilustración 26- Rango Bajo ([0.3,2])</td><td> 

Ilustración 27- Rango Alto ([3,1000])</td></tr>
</table>

Probabilidad de cambio	 **Ilustración 28-Probabilidad de cambio Baja (0.00001)**	 **Ilustración 29-Probabilidad de cambio Alta (0.95)**
Ruido Escáner	 **Ilustración 30- Ruido Escáner Bajo (10000)**	 **Ilustración 31- - Ruido Escáner Alto (10)**
Ruido Background	 **Ilustración 32- Ruido Background Bajo (0.0001)**	 **Ilustración 33- Ruido Background Alto (0.5)**

En los siguientes párrafos se explicará cada una de las variables y la razón de los resultados anteriores.

En el caso del parámetro de probabilidad de metilación cuanto más bajo sea el valor de esta variable más densidad se concentrará en torno a 0. Por el contrario, cuanta más probabilidad tenga de estar metilado el valor beta se concentrará cerca del uno, como es lógico. Esta prueba nos viene bien para darnos cuenta que nuestra implementación está bien realizada y que los gráficos empiezan a parecerse a los reales.

Por otro lado, respecto a los valores máximos del rango, tanto para el parámetro α (en sites metilados) como para el β (en sites demetilados), siendo este valor muy alto hace que los valores beta se concentren en los extremos. Esto es, si el valor máximo del rango para el α de los datos metilados es muy alto, los resultados de la distribución Beta se situarán muy cerca del uno y por el contrario, si valor máximo del rango para el β de los datos demetilados es muy alto, los resultados de la distribución Beta se situarán muy cerca del cero. No compensa tener datos cercanos a los extremos, por un lado porque, en los datos reales, hay escasos datos con valor igual a uno, y por otro lado, porque el tener datos cercanos al cero en nuestra implementación hace que la varianza de los datos Beta tienda a ser muy pequeña.

Por otra parte, una probabilidad de cambio alta hace que los datos sean inversos a los resultados deseados. Y en cuanto a un valor bajo hace que los valores beta queden en los extremos dejando los valores intermedios sin representación. En este caso, un valor bajo, pero no extremadamente bajo, sería una buena elección.

Por último, respecto al ruido del escáner, cuanto más pequeño es el parámetro, mayor es la varianza y, por tanto, mayor es el ruido. Por el contrario, en el background el ruido aumenta con el parámetro, ya que este representa directamente la variabilidad.

Después de realizar muchas pruebas se puede apreciar que los datos van cogiendo un parecido a los datos reales. A continuación se muestra en la Figura 34 los resultados finales de los datos creados junto con los datos reales.

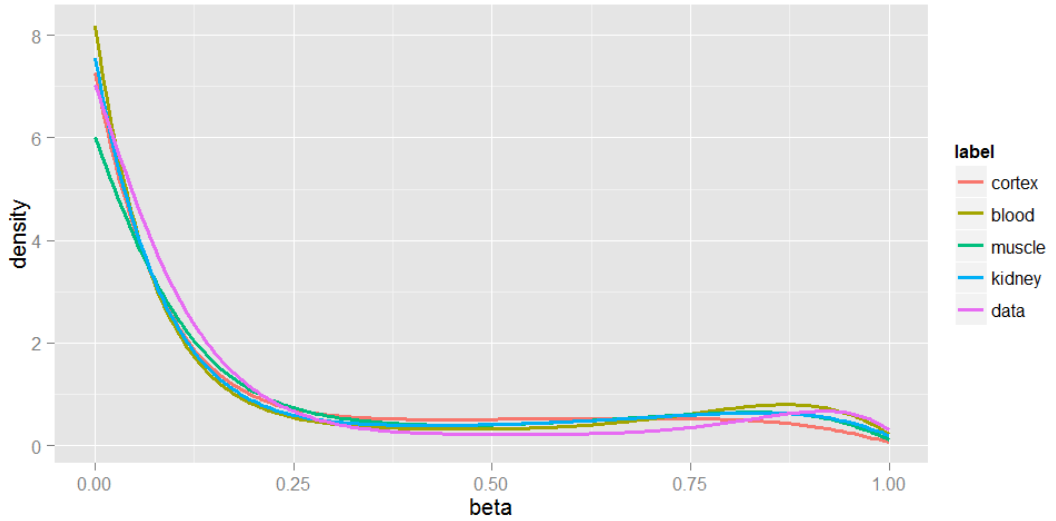


Ilustración 34- Distribución de los niveles de metilación de datos creados junto con los datos reales

De modo que se puede decir que hay unos ciertos rangos de valores para los parámetros que hacen que los datos creados se parezcan a los datos reales.

- Probabilidad de metilación=0.1-0.25
- Rango Alpha metilado=[3,8]-[3,15]
- Rango Beta demetilado=[3,3]-[3,40]
- Probabilidad de cambio=0.05-0.1
- Ruido Escáner=100-10000
- Ruido Background=0.5-0.01

Como bien se ha mencionado antes, los datos tiene una cierta relación entre la media y la varianza. Por ese motivo, se han representados gráficamente los datos creados en la Figura 35.

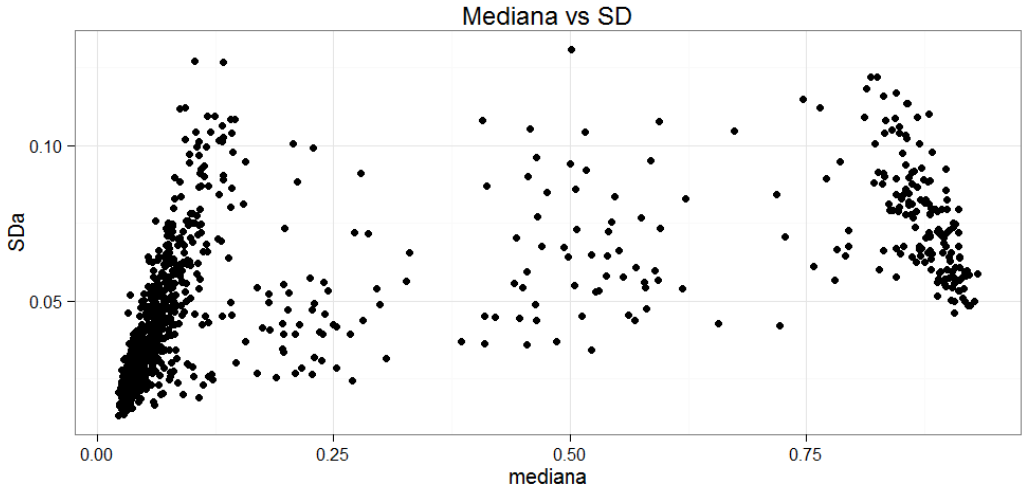


Ilustración 35- Relación entre la metilación media y la varianza

Comparando los resultados obtenidos con los datos reales se puede concluir que los resultados no son lo suficientemente buenos aunque empiezan a asemejarse. Sin embargo, hay que tener en cuenta la disminución de cantidad de datos, ya que la ejecución con una suma mayor de datos resultaría ser muy costosa.

Llama la atención la ausencia de valores próximos a 0. Una posible explicación de esto es que en los datos reales, los valores de 0 sean debido a problemas tales como un mal funcionamiento de la PCR, situación que el modelo no contempla.

2.5.IMPLEMENTACIÓN Y SU USO

Se han implementado una serie de funciones de R para imitar el comportamiento del sistema biológico. A continuación exploraremos el uso de cada una de ellas. El código de la implementación puede verse en el Anexo de este documento.

El objetivo de la primera función, "create_clone_model()" es crear para todos los individuos, un modelo general de muestra. Hace uso de los parámetros meth_prob, alpha_meth, beta_umeth, state_prob y num_clones, y devuelve una lista con los valores α y β para cada clon para un mismo site.

Como primer paso, se creara una célula o clon; a este clon lo llamaremos clon maestro. Luego se crean los clones restantes. El clon maestro tendrá mayor peso en la muestra y el resto dependerá de este clon, esto es, cambiará su estado según la probabilidad de modificación (state_prob).

Un dato fundamental para poder implementar esta primera función es la probabilidad de metilación (meth_prob) que, por defecto, vale 0.25. Así que, el clon maestro partiendo de esta probabilidad tomará aleatoriamente el valor metilado o demetilado; A continuación se generarán los valores α y β. En el caso de que este clon este metilado, entonces β=1, y por el contrario, si este clon esta demetilado, α=1. Por supuesto, es necesario fijar el otro parámetro. Para ello, se puede usar la desviación típica, pero esto implica resolver la siguiente ecuación:

$$\vartheta^2 = \frac{\alpha\,\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Con el propósito de mejorar la implementación y la eficiencia de los cálculos, la función toma como parámetro el rango en el cual pueden variar los parámetros y generar los valores de manera aleatoria uniforme. Esto es, habrá dos parámetros, "alpha_meth" que nos indicará el rango en el cual se mueve el valor de α de la distribución Beta cuando β=1 y "beta_unmeth", que nos dirá el rango en el cual se mueve el valor de β de la distribución Beta cuando α=1.

Igualmente, se lograrán los datos de los clones restantes, a pesar de que dependen del clon maestro y del parámetro de modificación, "state_prob". Mediante este último valor se sabrá si el estado de metilación del clon debe de ser modificado o no con respecto al clon maestro. Es decir, si el clon maestro no está metilado, este nuevo clon tendrá una probabilidad, por ejemplo, de 0.05 de estar metilado.

Para ilustrar mejor lo explicado, supongamos que tenemos una muestra con 4 clones. El clon maestro tendrá una probabilidad de 0.25 de estar metilado. Digamos que está demetilado, de manera que ya sabemos que α=1. Suponiendo que el rango para determinar el valor β es [3,35] beta será:

$$\beta = min + num.aleatorio * (max - min) = 3 + num.aleatorio * (35 - 3)$$

De esta manera logramos que β sea, por ejemplo, β=24.75071.

Suponiendo que el rango para determinar el valor α es [3,10], los clones restantes obtendrán, por ejemplo, los siguientes resultados:

	α	β
Clon Maestro	1	24.75071
Clon1	1	21.70497
Clon2	1	12.35452
Clon3	5.23	1

En definitiva, será la lista compuesta por los valores α y β la que se guardará para las próximas funciones.

```
> prop<-c(80,2,8,20)

> cc<-create_clone_model(prop,meth_prob=0.25, alpha_meth=c(3,10),
beta_umeth=c(3,35), state_prob=0.05)
> cc
        [,1]     [,2]     [,3]     [,4]
[1,]  1.0000 1.000000  1.00000  5.763584
[2,] 24.0741 21.521896 12.29756 1.000000
```

Con esta función tenemos la distribución Beta que define cada clon. Ademas de esto necesitamos los pesos asociados a cada tipo celular. Estos pesos los podemos fijar a mano. Como alternativa se ha implementado la función RunProp(), que genera estos pesos de manera aleatoria siguiendo una distribución exponencial.

```
> prop<-RunProp(4)
> prop
[1] 74.552337  16.116853  7.634299  1.696511
```

Llegados a este punto, ya tenemos una muestra compuesta por distintos tipos de células, cada uno con su proporción o peso correspondiente.

La función, “create_tissue_model()” determina los parámetros de la Direchlet que simulará los porcentajes de la muestra.

Las proporciones se logran mediante la siguiente función:

$$ParamDirichlet = \frac{\alpha_T * Proporciones}{\sum(Proporciones)}$$

Donde, α_T es el valor que determina la incertidumbre respecto a los pesos. Si α_T tiene un valor grande, el muestreo dará como resultado proporciones muy similares a las canónicas.

De modo que esta función devolverá un modelo de tejido o “tissue model”. Será una lista compuesta por la lista con los valores α y β y las proporciones correspondientes como parámetros de Dirichlet.

```
> prop
[1] 74.552337  16.116853  7.634299  1.696511

> cc
         [,1]      [,2]     [,3]     [,4]
[1,]  1.0000 1.000000  1.00000  5.763584
[2,] 24.0741 21.521896 12.29756 1.000000

> ct<-create_tissue_model(cc,prop,alpha_Tot=(100+ runif(1)*10))
> ct
$clone
         [,1]      [,2]     [,3]     [,4]
[1,]  1.0000 1.000000  1.00000  5.763584
[2,] 24.0741 21.521896 12.29756 1.000000

$dirParm
[1] 80.449722  17.391760  8.238202 1.830712
```

La siguiente función, “sample_tissue()”, tiene como fin generar los valores beta, esto es, el ratio de metilación. Para ello, serán necesarios los parámetros α y β de cada clon y los parámetros de la distribución Dirichlet obtenidos anteriormente.

El muestreo de la Dirichlet se logra fácilmente haciendo uso del muestreo de la gamma ya implementada en R, “rgamma()”.

Cada clon o modelo de clon tiene guardados los parámetros α y β. Estos dos parámetros son fundamentales para poder muestrear el valor beta. Para esto, es necesario utilizar la función implementada en R, “rbeta()”.

Luego, se multiplicarán cada valor beta muestreado con el valor de la distribución de Dirichlet o peso que le corresponde. Finalmente, se sumarán para obtener el valor beta correspondiente a cada individuo. De igual manera se lograrán los valores para todos los individuos.

Todas las posiciones CpG de un individuo generan lo que llamamos un modelo de individuo o “individual_model”. En este punto, tendremos los valores beta correspondiente a una posición CpG en concreto y un individuo concreto.

Con todas las funciones descritas anteriormente se han integrado en una única función que simule el comportamiento de una sola posición CpG, “BetaForSite()”.

```
> BetaForSite(proportions=prop,alpha_Tot=100+runif(1)*10,meth_prob=0.25,
alpha_meth=c(3,10),beta_umeth=c(3,35),state_prob=0.05,num_inds=10,bw=0.05,
trim=0.05)$sample

 [1] 0.023472818 0.019656523 0.040805360 0.016028109 0.018486753 0.006132417
0.003081451 0.022964162 0.101401790
[10] 0.043861428
```

Veamos como genera los datos para 10 individuos y 5 sites.

```
> betas<-sapply(1:(num_sites=5),FUN=function(x){BetaForSite(proportions=prop,
alpha_Tot=100+runif(1)*10,meth_prob=0.25,
alpha_meth=c(3,10),beta_umeth=c(3,35),
state_prob=0.05,num_inds=10,bw=0.05,trim=0.05)$sample})
> betas
             [,1]         [,2]       [,3]         [,4]      [,5]
 [1,] 0.04240956 0.019839207 0.03187185 0.022708699 0.5644902
 [2,] 0.05813940 0.004249576 0.03886510 0.024926000 0.7572385
 [3,] 0.02285318 0.029985205 0.03632710 0.026698996 0.7535252
 [4,] 0.09578409 0.039296876 0.01463937 0.046932174 0.7582167
 [5,] 0.07812203 0.040533237 0.07134558 0.059315702 0.4720635
 [6,] 0.03345122 0.024390595 0.01921310 0.057952914 0.7111841
 [7,] 0.04803599 0.014731368 0.01292371 0.012008089 0.5991788
 [8,] 0.05198711 0.029407725 0.03876456 0.009985741 0.7527314
 [9,] 0.06060519 0.036423618 0.04113257 0.020146732 0.6656677
[10,] 0.05314612 0.041125401 0.03277315 0.021993149 0.7493924
```

Con el modelo de individuo la siguiente función “sample_individual()”, tiene como objetivo muestrear la distribución binomial para conseguir los valores de metilación correspondientes a un individuo.

```
> Site1<-sample_tissue(ct, num_inds=10)
> Site1
 [1] 0.07387439 0.08303537 0.15354532 0.08824569 0.09313568 0.05195582
0.02509461 0.01083461 0.04863287 0.06257515
> Site2<-sample_tissue(ct, num_inds=10)
> Site2
 [1] 0.08238629 0.06128397 0.09930262 0.15858170 0.04430199 0.08699879
0.06068725 0.09286658 0.20664737 0.04624717
```

Se le aplicará una distribución binomial en donde el parámetro p de dicha binomial es la β obtenida, pero nos falta el parámetro N.

Como hemos comentado antes, usaremos valores grandes de N para reducir la variabilidad. De modo que, obtendremos la n aleatoriamente mediante una distribución normal, con μ=10000 y σ=100. Estos valores por defecto pueden ser modificados por el usuario.

Por último, para cada posición CpG se logrará la cantidad de datos metilados siguiendo la distribución binomial con la probabilidad de éxito igual al nivel de metilación real, esto es, el modelo de individuo creado anteriormente y las muestras N. Los datos no metilados es la diferencia entre la N y los datos metilados, de modo, U=N-M.

```
> samInd<-sample_individual(betas[1,], num_sites=5, N=createMol(num_sites=5))
> samInd
  methylated unmethylated
1        407          9581
2        198          9777
3        296          9718
4        238          9836
5       5594          4433
```

Durante todo el proceso químico y tecnológico es probable que sucedan diversos errores. Para representar esos problemas, se ha implementado la siguiente función, “analyse_sample()”.

Para atender a la tasa de errores que se puedan generar, como problemas de hibridación, o problemas de una intensidad de background elevada debido a un mal lavado, es necesario introducir cierto ruido en los datos.

Este ruido se añade en la función “FinalBetas()” que toma como entrada los valores beta teóricos y genera las intensidades por cada individuo y para todos los sites, y estima el valor veta final.

El valor beta final se obtiene como la relación entre la intensidad metilada y la intensidad global. Basándonos en esto, podremos calcular el valor beta mediante la siguiente función:

$$beta = \frac{I_M}{I_M + I_U + offset} \qquad offset = 100$$

En cuanto al valor de desplazamiento, “offset”, teniendo en cuenta que los valores de intensidad son mayores que 10000, un valor como 100 es relativamente pequeño y tiene un efecto despreciable sobre el valor Beta; si el offset es 0 no tiene efecto alguno.

```
> head(betas)
           [,1]        [,2]       [,3]       [,4]      [,5]
[1,] 0.04240956 0.019839207 0.03187185 0.02270870 0.5644902
[2,] 0.05813940 0.004249576 0.03886510 0.02492600 0.7572385
[3,] 0.02285318 0.029985205 0.03632710 0.02669900 0.7535252
[4,] 0.09578409 0.039296876 0.01463937 0.04693217 0.7582167
[5,] 0.07812203 0.040533237 0.07134558 0.05931570 0.4720635
[6,] 0.03345122 0.024390595 0.01921310 0.05795291 0.7111841

> finBeta<-FinalBetas(betas, scaner_k=10000, zG=0.01)
> head(finBeta)
[1] 0.04603453 0.05977368 0.02029899 0.09420468 0.08000783 0.03085491
```

Por último, se ha generado la función general, nombrada como “mainFunction()”. Por un lado consigue los valores beta, de site en site para todos los individuos, para crear un modelo de individuo. Y luego, muestrea individuo a individuo, creando un modelo de muestra real. Todo ello será representado matricialmente o gráficamente mediante la función “plot.sample.density()”.

En todo lo anterior no hemos tenido en cuenta la imprintación. En un porcentaje pequeño de los genes la metilación heredada de cada progenitor es distinta. Para poder simular este proceso se ha implementado la siguiente función, “mainFunctionImprented()” que se basa en llamar a la función anterior dos veces y se realiza una media de los dos resultados.

Por otro lado, para el fácil uso de los usuarios se ha creado una aplicación Web. Es una aplicación muy sencilla de utilizar implementada mediante la herramienta Shiny.

Para empezar, es necesario instalar el paquete “Shiny”. Todo app de Shiny sigue la misma estructura a la hora de implementarla. Es necesario crear dos archivos, ui.R y server.R. Estos dos archivos se deberán guardar en un mismo directorio para después poder ejecutarlo.

Dentro del archivo ui.R, se diseña la interfaz gráfica, mientras que en server.R, está la llamada de la función general de nuestro programa. Es necesario tener en cuenta los parámetros necesarios en cada uno de los archivos.

Finalmente para poder ejecutar el programa se debe de realizar la siguiente llamada:

$$runApp("nombreDelDirectorio")$$

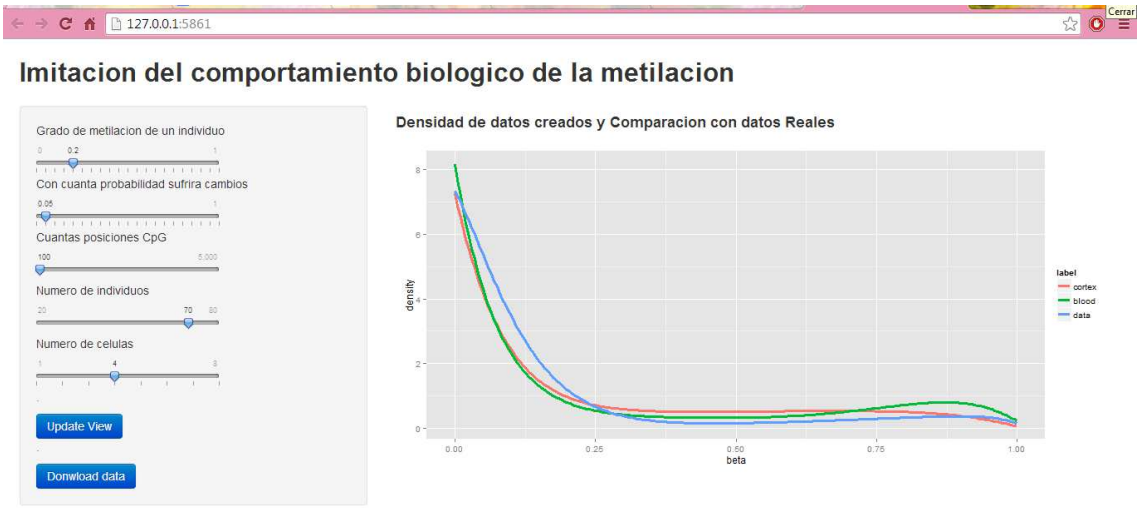


Ilustración 36- Ejemplo de uso de la aplicación

El uso de nuestra aplicación es muy sencilla en la parte de la izquierda (ver Figura 36) se deben de indicar los parámetros que se desean y en la derecha se imprimirán los dos gráficos demostrando el nivel de metilación acorde a los parámetros establecidos. Hay dos botones; con “Update View” se pueden editar los parámetros y actualizar el gráfico, y con “Download data” se podrán descargar los datos que se han creado.

3.CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado un nuevo algoritmo que imita el comportamiento biológico de la metilación basado en la estadística y la probabilidad, el cual se fundamenta el análisis y la exploración de datos reales detallada.

Se ha desarrollado un sistema utilizable por cualquier usuario que llega a imitar el comportamiento biológico. Se han tratado de modelar tanto el proceso biológico humano como los procesos tecnológicos necesarios, tales como la conversión del bisulfito, la hibridación o el escaneo.

Se han realizado pruebas iterativas para ajustar dinámicamente los parámetros, de tal forma que el resultado se asemeje a los datos reales.

Los resultados obtenidos son bastante satisfactorios, sin embargo aún hay cuestiones que requieren más estudio. Las distribuciones globales obtenidas se asemejan bastante a la realidad. Sin embargo, en el caso de la representación media vs varianza queda de manifiesto que aún los resultados no son del todo realistas.

Otra de las cuestiones que quedaron pendientes es el estudio de las fuentes de error sistemático, tales como problemas con la PCR o la conversión con bisulfito.

Como trabajo futuro, además del refinamiento del modelo, cabe destacar la necesidad de un estudio más exhaustivo de los datos reales, a fin de tratar de estimar valores razonables para los parámetros del modelo.

Por último, no hay que olvidar que el objetivo es crear datos artificiales que sirvan para probar las herramientas diseñadas para trabajar con datos reales. Por este motivo, conviene generar los datos en un formato compatible con esas herramientas, tales como los objetos de tipo LumiBatch del paquete lumi de bioconductor. En relación con esto, cabe destacar que estos objetos, además de los datos, contienen anotaciones de los mismos, que se podrán utilizar para indicar que posiciones muestra diferencias entre grupos, etc.

BIBLIOGRAFIA

[1] Keith Wilson, John M. Walker; Principles and Techniques of Biochemistry and Molecular Biology, 7th Edition, University of Cambridge (2010)

[2] Zachary D. Smith & Alexander Meissner; DNA methylation: roles in mammalian development (2013)

[3] Yang SY, Yang XL, Yao LF, Wang HB, Sun CK, Effect of CpG methylation on DNA binding protein: Molecular dynamics simulations of the homeodomain PITX2 bound to the methylated DNA (2011)

[4] Hao Zheng, Hongwei Wu, Jinping Li and Shi-Wen Jiang, CpGIMethPred: computational model for predicting methylation status of CpG islands in human genome (2011)

[5] Zhou X, Li Z, Dai Z, Zou X, Prediction of methylation CpGs and their methylation degrees in human DNA sequences (2011)

[6] Christoph Bock, Analysing and interpreting DNA methylation data (2012)

[7] Elías Canetti, Epigenética: una explicación de las enfermedades hereditarias (2005)

[8] Pan Du, Xiao Zhang, Chiang-Ching Huang, Nadereh Jafari, Warren A Kibbe, Lifang Hou, Simon M Lin, Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis (2010)

[9] Kirti Laurila, Bodil Oster, Claus L Andersen, Philippe Lamy, Olli Yli-Harja, and Carsten Wiuf, A beta-mixture model for dimensionality reduction, sample classification and analysis. BMC Bioinformatics (2011)

[10] Bibikova M, Barnes B, Tsan C, Ho V, Klotzle B, Le JM, Delano D, Zhang L, Schroth GP, Gunderson KL, Fan JB, Shen R, High density dna methylation array with single cpg site resolution (2011)

[11] A. Meissner, A. Gnirke, G.W. Bell, B. Ramsahoye, E.S. Lander, R. Jaenisch, Reduced representation bisulfite sequencing for comparative high-resolution DNA methylation analysis (2005)

[12] LINDA VAN Speybroeck, Desde Epigénesis a Epigenética El caso de la CH Waddington (2002)

[13] J G Herman, J R Graff, S Myöhänen, B D Nelkin, and S B Baylin, Methylation-specific PCR: a novel PCR assay for methylation status of CpG islands (1996)

[14] Singal R, Ginder GD, DNA methylation. Blood (1999),

[15] Herman JG, Baylin SB, Gene silencing in cancer in association with promoter hypermethylation. (2003)

AGRADECIMIENTOS

Este Proyecto de Fin de Máster no hubiera sido posible sin la ayuda y el apoyo de muchas personas, a las que me gustaría dar mis agradecimientos.

Sin lugar a duda, la persona fundamental en este trabajo ha sido mi tutor, Borja Calvo, quien me ha guiado en el mundo de la investigación y con quien he aprendido mucho. Tengo que darle las gracias por todos los consejos, orientación y recomendaciones durante todo el proceso.

Quiero agradecer también a Naiara G. Bediaga por toda la ayuda empleada, por sus sugerencias y discusiones durante la tesis.

Finalmente, gracias también a familiares y amigos por vuestro apoyo y cariño.

ANEXOS

```
#INPRINTAZIOA
mainFunctionImprented<-
function(num_sites=1000,num_rep=4,proportions=RunProp(num_rep+1),alpha_Tot=(100+runi
f(1)*10),
     meth_prob=0.25,alpha_meth=c(3,4),beta_umeth=c(3,17),state_prob=0.05,num_inds=100,
     bw=num_inds^(-2/5),trim=0.05, scaner_k=10000, zG=0.01){

  print(proportions)
  if(num_sites==1){
    inpr<-0
    betas_inpr<- NA
  }
  else{
    inpr<-round(num_sites/250)+1
    a<-mainFunction(inpr,num_rep,proportions,alpha_Tot,meth_prob,alpha_meth,
beta_umeth, state_prob,num_inds,bw, scaner_k, zG)$sample
    b<-mainFunction(inpr,num_rep,proportions,alpha_Tot,meth_prob,alpha_meth,
beta_umeth, state_prob,num_inds,bw, scaner_k, zG)$sample
    betas_inpr<- a*0.5+b*0.5
  }

  betas<-mainFunction(num_sites-inpr,num_rep,proportions,alpha_Tot,meth_prob,
alpha_meth, beta_umeth,state_prob,num_inds,bw, scaner_k, zG)$sample
  list(plot=plot.sample.density(c(betas,betas_inpr),bw=0.01,alpha=0.5),
sampleBeta=t(betas),sampleInpr=t(betas_inpr))

}



mainFunction<-
function(num_sites=1000,num_rep=4,proportions=RunProp(num_rep+1),alpha_Tot=(100+runi
f(1)*10),

meth_prob=0.25,alpha_meth=c(3,4),beta_umeth=c(3,17),state_prob=0.05,num_inds=100,bw=
num_inds^(-2/5)
                  , scaner_k=10000, zG=0.01, trim=0.05){

  betas<-sapply(1:(num_sites),FUN=function(x){BetaForSite(proportions,alpha_Tot,
meth_prob, alpha_meth,beta_umeth,state_prob,num_inds,bw)$sample})
  beta.datos<-lapply(1:(num_inds),FUN=function(x){FinalBetas(betas[x,], scaner_k,zG)})
  beta.datos<-matrix(unlist(beta.datos), nrow=num_sites, ncol=num_inds)
  list(plot=plot.sample.density(beta.datos,bw=0.05,alpha=0.5), sample=beta.datos)

}
```

```
FinalBetas<-function(beta, scaner_k=10000, zG=0.01){
 num_sites<-length(beta)
 sample_model<-sample_individual(beta,num_sites)

 #                                                                                    beta<-
sample_model$methylated/(sample_model$methylated+sample_model$unmethylated+100)
 beta<-
sample_model$methylated/(sample_model$methylated+sample_model$unmethylated)

 intentsitateak<-analyse_sample(sample_model, num_sites, scaner_k, zG)
 #                                                                                    beta<-
intentsitateak$methylated/(intentsitateak$methylated+intentsitateak$unmethylated+100)
beta<-intentsitateak$methylated/(intentsitateak$methylated+intentsitateak$unmethylated)
 return(beta)
}
##
BetaForSite<-
function(proportions=RunProp(num_rep+1),alpha_Tot=100+runif(1)*10,meth_prob=0.25,alph
a_meth=c(3,4),beta_umeth=c(3,17),state_prob=0.05,num_inds=100,bw=num_inds^(-
2/5),trim=0.05){

 model<-create_clone_model(proportions,meth_prob,alpha_meth, beta_umeth,state_prob)
 tm<-create_tissue_model(model,proportions, alpha_Tot)
 sample<-sample_tissue(tm, num_inds)
 list(plot=plot.site.density(sample,trim=trim,bw=bw),sample=sample)
}

# INPLEMENTATZEKO ----------------------------------------------------------

create_clone_model<-
function(weights,meth_prob=0.25,alpha_meth=c(3,4),beta_umeth=c(3,17), state_prob=0.05){

 sample_cell_type<-function(x){
  if(runif(1)<state_prob){
   if(clon_maestro[1]==1){
    b=1
    a=alpha_meth[1] + runif(1)*(alpha_meth[2]-alpha_meth[1])
   }
   else{
    a=1
    b=beta_umeth[1] + runif(1)*(beta_umeth[2]-beta_umeth[1])
   }
  }else{
   if(clon_maestro[1]==1){
    b=beta_umeth[1] + runif(1)*(beta_umeth[2]-beta_umeth[1])
   }
   else{
    a=alpha_meth[1] + runif(1)*(alpha_meth[2]-alpha_meth[1])
   }
  }
  list(a,b)
 }
```

```
 if(runif(1)>meth_prob){
   a=1
   b=beta_umeth[1] + runif(1)*(beta_umeth[2]-beta_umeth[1])
 }else{
   b=1
   a=alpha_meth[1] + runif(1)*(alpha_meth[2]-alpha_meth[1])
 }
 clon_maestro<-c(a,b)
 model<-matrix(c(clon_maestro,unlist(sapply(1:(length(weights)-
1),FUN=sample_cell_type))),nrow=2)

}


RunProp<-function(VectorSize){
 if(VectorSize>4){
   baloreak<-beta.parms.from.quantiles(c(0.75,0.8),plot=F)
   nagusia<-rbeta(1,baloreak$a,baloreak$b)*100
   Scalar=100-nagusia
   RandomVector=round(rbeta(VectorSize-2,1,1)*20)+1
   aux<-round(runif(1)*nagusia)
   emaitza<-c(aux,nagusia-aux)
 }
 else{
 baloreak<-beta.parms.from.quantiles(c(0.7,0.8),plot=F)
 nagusia<-rbeta(1,baloreak$a,baloreak$b)*100
 Scalar=100-nagusia
 RandomVector=round(rbeta(VectorSize-1,1,1)*20)+1
 emaitza<-nagusia
 }
 RandomVectorSum=sum(RandomVector)
 RandomVector=lapply(RandomVector,FUN=function(x){Scalar*x/RandomVectorSum})
 return(c(unlist(RandomVector),emaitza))
}

create_tissue_model<-function
(clone_list,proportions=RunProp(num_rep+1),alpha_Tot=(100+ runif(1)*10)){
 w<-sum(proportions)
 # Dirichlet distribuzioaren parametroak kalkulatu behar dira desbiderazioa alpha_tot
 #   parametroaren bidez pasatutakoa izateko
 WE<-alpha_Tot*proportions/w
 tm<-list(clone=clone_list, dirParm=WE)
 return(tm)
}

## Funtzio hau Dirichlet distribuzioa lagintzeko erabili ahal da. Hauxe klon ereduak nahasteko
erabiliko dugu
rdir = function(n, alpha_vector) {
 r<-sapply(1:length(alpha_vector),FUN=function(x){rgamma(n,alpha_vector[x],1)})
 if (n==1) r/sum(r) else r/rowSums(r)
}
```

```
sample_tissue<-function(tissue_model, num_inds=100){
 model<-tissue_model$clone
 weights<-rdir(1,tissue_model$dirParm)
 s<-function(x){
  sum(apply(model,MARGIN=2,FUN=function(x){rbeta(1,x[1],x[2])})*weights)
 }
 sample<-sapply(1:num_inds,FUN=s)
}

datuakLortu <-function (num_sites){
 mu=10000
 sigma=mu/100
 return(round(rnorm(num_sites,mu,sigma)))
}

sample_individual <-function (individual_model,num_sites,N=datuakLortu(num_sites)){

 ##Indibiduo batetik ateratako laginak
 pr<-individual_model
 mth<-sapply(1:num_sites,FUN=function(x){rbinom(n=1,size=N[x],prob=pr[x])})
 sample_model<-data.frame(methylated=mth,unmethylated=N-mth)
 return(sample_model)
}

analyse_sample<-function (sample_model,bis_conv=runif(1)*0.02, scaner_k=20, backP=0,
ZarataGauss=0){
 #BISULFITO---
 if(bis_conv!=1){
 sample_model$methylated<-
sample_model$methylated+sample_model$unmethylated*bis_conv
 sample_model$unmethylated<-(1-bis_conv)*sample_model$unmethylated
 }
 #SCANER---
 meth<-sapply(1:num_sites,FUN=function(x){

rnorm(n=1,mean=sample_model$methylated[x],sd=sample_model$methylated[x]/scaner_k)})
 unmeth<-sapply(1:num_sites,FUN=function(x){

rnorm(n=1,mean=sample_model$unmethylated[x],sd=sample_model$unmethylated[x]/scane
r_k)})
 intentsitateak<-data.frame(methylated=meth,unmethylated=unmeth)

 #BACKGROUND---
 intentsitateak$methylated<-intentsitateak$methylated+ rnorm(n=1,mean=backP,sd=zG)
 intentsitateak$unmethylated<-intentsitateak$unmethylated+ rnorm(n=1,mean=backP,sd=zG)
 return(intentsitateak)
}
```

```
# PLOTS: Densities of beta-values ----------------------------------------

plot.sample.density<-function (sample,elph=T,limit=50000,bw=min(length(sample),limit)^(-
2/5),alpha=2000/min(length(sample)),normalize=F){
 sample<-as.vector(sample)
 if (length(sample)>limit) sample<-sample[runif(limit,1,length(sample))]

 #scale to 0-1
 if (normalize){
  m<-min(sample)
  M<-max(sample)
  sample<-(sample-m)/(M-m)
 }
 beta<-seq(0,1,0.005)
 dens<-csk.eval(samples=sample,x=beta,b=bw,kernel="chen99",mu=1)
 if (normalize) {
  df<-data.frame(beta=dens$x*(M-m)+m,density=dens$y)
 }else{
  df<-data.frame(beta=dens$x,density=dens$y)
 }
 plot<-ggplot(df,aes(x=beta,y=density))
 if (elph){
  dg<-max(dens$y)/10
  df2<-data.frame(x=sample,y=-dg*1.5)
  plot<-plot +
geom_point(data=df2,mapping=aes(x=x,y=y),position=position_jitter(height=dg),col="black",al
pha=alpha)
 }
 plot + geom_line(size=1.1)
}




plot.site.density<-function(sample,trim=0,bw=length(sample)^(-2/5)){
 if (mode(sample)=="list"){
  bt<-seq(0,1,0.005)
  df<-data.frame()
  for (i in 1:length(sample)){
   print(sample[[i]])
   s<-sort(as.vector(sample[[i]]))
   if (trim>0){
    if(trim>=1){
     warning("The trim value should be lower than 1")
    }else{
     l<-length(s)
     i<-l*trim
     j<-l-i
     s<-s[i:j]
    }
   }
   dens<-csk.eval(samples=s,x=bt,b=bw,kernel="vitale",mu=1)
```

```
    df<-rbind(df,data.frame(beta=dens$x,density=dens$y,label=names(sample)[i]))
   }
   plot<-ggplot(df,aes(x=beta,y=density,col=label))
   plot + geom_line(size=1.1)
  }else{
   sample<-sort(as.vector(sample))
   if (trim>0){
    if(trim>=1){
     warning("The trim value should be lower than 1")
    }else{
     l<-length(sample)
     i<-l*trim
     j<-l-i
     sample<-sample[i:j]
    }
   }
   bt<-seq(0,1,0.005)
   dens<-csk.eval(samples=sample,x=bt,b=bw,kernel="vitale",mu=1)
   df<-data.frame(beta=dens$x,density=dens$y)
   plot<-ggplot(df,aes(x=beta,y=density))
   dg<-max(dens$y)/10
   df2<-data.frame(x=sample,y=-dg*1.5)
   plot<-plot                                                                    +
geom_point(data=df2,mapping=aes(x=x,y=y),position=position_jitter(height=dg),col="black",al
pha=1)
   plot + geom_line(size=1.1)
  }
}
```