**PEDECIBA Informática**
**Instituto de Computación – Facultad de Ingeniería**
**Universidad de la República**
**Montevideo, Uruguay**

# Tesis de Maestría
## en Informática

# Análisis del Muestreo Gibbs para Detección de Motivos en Secuencias Biológicas

**Laura Angelone**

**Octubre 2005**

**Orientador de Tesis:** **Dra. Elizabeth Tapia Paredes y MsC. María Urquhart**

**Supervisor:** **MsC. María Urquhart**

Análisis del muestreo Gibbs para detección
de motivos en alineaciones múltiples
Laura Angelone

ISSN 0797-6410
Tesis de Maestría en Informática
Reporte Técnico RT
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay, Octubre de 2005

## RESUMEN

El reconocimiento de patrones comunes o *motivos* en la evolución, disposición estructural y funcionalidad biológica de un conjunto de secuencias biológicas (ADN o proteínas) es aún hoy un desafío importante en Biología Computacional. El problema requiere la determinación simultánea de la composición y ubicación de los motivos comunes a partir del conjunto de secuencias afectadas por ruido de evolución y desalineadas. De acuerdo a los trabajos de Ming Li *et al*. [44][45], la determinación de una solución exacta es un problema NP completo y por lo tanto la formulación de soluciones aproximadas es de fundamental interés. En particular, el modelado estadístico de secuencias mediante modelos ocultos de Markov (HMM) o mediante Muestreo Gibbs permite el diseño de aproximaciones biológicamente significativas sujeto a la disponibilidad de un número adecuado y variado de secuencias. Estas restricciones son especialmente limitantes en el caso de modelos HMM pero salvable en muestreo Gibbs admitiendo una carga computacional ligeramente mayor. A diferencia del modelado HMM, el cual asume una determinada estructura para el proceso de generación de datos, el muestreo Gibbs intenta aproximar la distribución de probabilidad que rige a los datos bajo estudio en un proceso iterativo caracterizado por una gran simplicidad algorítmica.
En esta tesis se analizan tanto los aspectos teóricos como prácticos que rigen el muestreo Gibbs para el problema de detección de motivos. Los resultados de este análisis se encuentran en la implementación de un software específico, su aplicación a la determinación de motivos en familias de secuencias de proteínas muy divergentes encuadradas en el Proyecto "Caracterización de factores basales de trascripción en parásitos protozoarios", Serra *et al.*[56], y su comparación con los programas de uso libre Gibbs Sampling[32] y MEME[5] .

---

PALABRAS CLAVE: Reconocimiento de Patrones, Biología Computacional, Alineación Múltiple de Secuencias biológicas, Muestreo Gibbs.

Key words: Pattern Recognition, Computational Biology, Multiple Alignment of Sequences, Gibbs sampling.

# Indice

Montevideo, 3 de Octubre de 2005

Señores
FACULTAD DE INGENIERIA
BIBLIOTECA DEL INSTITUTO DE COMPUTACION Y
DEL AREA INFORMATICA DEL PEDECIBA

El autor autoriza a la Biblioteca del Instituto de Computación de la Facultad de Ingeniería y del Area Informática del PEDECIBA la reproducción total o parcial de este documento, y su difusión a través de cualquier medio, con la debida cita de reconocimiento de la autoría y con fines de investigación, docencia e institucionales.

_____

Ing. Laura Mónica Angelone
DNI 13255394
España 3425
Rosario - Santa Fe
República Argentina
Teléfono : 054-0341-4825984

## Financiación

Los estudios de esta Maestría han sido financiados con los aportes de las siguientes instituciones:

- ✓ Red Iberoamericana de Bioinformática (RIB) Proyecto 2003/2006,

- ✓ Fondo para el Mejoramiento de la Calidad Universitaria (FOMEC) Contrato UNR- 614-P,

- ✓ Agencia Nacional de Promoción Tecnológica y Científica de la República Argentina. Red para la Promoción de las Tecnologías de la Información y las Comunicaciones (ProTIC) Proyecto PAV2003-00127-00000 – Área de Vacancia "Tecnología de la Información y las Comunicaciones".

## Agradecimientos

Mi agradecimiento a Marita por haberme dado la oportunidad de realizar esta Maestría, y por su inmensa paciencia. A Elizabeth por haber depositado en mí su confianza y sus conocimientos incondicionalmente. A Silvia Di Marco por destrabarme en la Estadística. A Sergio Geninatti por su aporte invalorable. A Leonardo Ornella por ayudarme a comprobar que mi programa funciona correctamente desde un punto de vista biológico, seleccionando los diferentes grupos de secuencias para las pruebas y analizando los resultados. Al Dr. Esteban Serra que me ha permitido ingresar en este fascinante mundo de la Bioinformática. Y en especial a mi familia que tan estoicamente ha soportado mis ausencias.

**Para César, Marco y Celina**

# Capítulo 1: Introducción

*¿Y si tiene mi apariencia física y tu cerebro?*
*Bernard Shaw a Isadora Duncan*

## 1.1 Descripción general

La Bioinformática es un nuevo campo que ha nacido de la necesidad de altos requerimientos de recursos computacionales para organizar, analizar y almacenar información biológica con la finalidad de responder preguntas complejas en Biología [41]. Bioinformática es un área de investigación multidisciplinaria, donde se relaciona Biología molecular, Computación y Estadística. La misma fue impulsada por el enigma del genoma humano y la promesa de una nueva era en la cual la investigación genómica pueda ayudar a mejorar la condición y calidad de vida. En este nuevo escenario se dispone de grandes cantidades de datos: Genomas completos, información contenida en bases de datos públicas, ambiciosos proyectos de estudios masivos de interacción entre proteínas, y una variedad de software a medida. En este contexto se plantea un cambio de paradigma, un cambio en la mentalidad, una reorientación de las herramientas informáticas y de las estrategias para procesar datos biológicos.

Al comienzo de la revolución genómica, el concepto de Bioinformática se refería sólo a la creación y mantenimiento de bases de datos donde almacenar información biológica, tales como secuencias de nucleótidos y aminoácidos. Luego toda esa información debía ser combinada para formar una idea lógica de las actividades celulares moleculares, de tal manera que los investigadores pudieran estudiar cómo estas actividades se veían alteradas en estados de una enfermedad. De allí viene el surgimiento de la Bioinformática, en ayuda del investigador para analizar e interpretar datos biológicos. Tradicionalmente los métodos de laboratorio para el estudio de las moléculas son largos y pueden omitir alguna información importante. Como resultado de esto, los biólogos moleculares comenzaron a requerir métodos estadísticos capaces de analizar una gran cantidad de datos en tiempos más cortos y programas computacionales que implementaran dichos métodos. De esta manera aparecen desarrollos de nuevos algoritmos para vincular partes de este enorme conjunto de datos, para localizar un gen dentro de una secuencia, para predecir estructuras o funciones de proteínas, para

agrupar secuencias de proteínas en familias relacionadas. Sin embargo, y a pesar que mucho se ha logrado, aún sigue habiendo una variedad de desafíos importantes en Bioinformática.

Hoy en día, luego que una molécula se secuencia mediante métodos de laboratorio, una práctica básica para recabar información para su análisis procede de la comparación de la misma con datos procedentes de diversas bases de datos. La idea es buscar homologías frente a otras secuencias ya secuenciadas. La homología entre secuencias, además de sugerir un mismo origen filogenético, frecuentemente indica una correlación funcional de ambas proteínas. De este modo, se puede tener una idea de la función de una nueva secuencia si en la base de datos de proteínas se encuentra otra secuencia homóloga cuya función sea conocida. Para detectar dichas homologías entre secuencias completas de proteínas se utilizan métodos ya tradicionales como BLAST o FASTA. Sin embargo, la búsqueda de estas similitudes no es tan simple como podría parecer en un principio, es necesario tener en cuenta la posible existencia de variaciones como la aparición de eliminaciones o inserciones producidas por mutaciones evolutivas. Uno de los mecanismos ampliamente utilizados para esta búsqueda es la comparación mediante la alineación de secuencias. El principal problema en la comparación de secuencias consiste en encontrar todas las zonas de similitud significativa entre secuencias y determinar que es significativo a la hora del análisis biológico de las mismas. Las diferencias pueden ser invaluables a la hora de determinar, por ejemplo, porque dos individuos tienen diferente propensión a una determinada enfermedad.

El objetivo algorítmico de alinear 2 secuencias es encontrar la posición relativa de ambas en las que se produzca el mayor número de coincidencias entre sus componentes. Este número de coincidencias representa la valoración de su parecido. Este proceso revela en dónde las secuencias son iguales y en dónde no lo son. Mientras que el objetivo biológico de la comparación es inferir relaciones estructurales, funcionales o evolutivas entre las secuencias. La falta de semejanza se da como resultado del mecanismo de la evolución; moléculas que comparten un antepasado común no son precisamente iguales pero heredan muchas similitudes en la estructura primaria de su antepasado.

Una vez que se ha identificado un grupo de secuencias relacionadas a la secuencia bajo estudio, por ejemplo con BLAST, suele ser de utilidad determinar si hay regiones en dicho grupo que se conservan más que otras. O tal vez puede darse el caso que la secuencia analizada carece de similitud directa con las secuencias de la base de datos. En este caso, el análisis puede ser realizado por comparación con familias de secuencias ya conocidas. Para lograr este propósito es imprescindible lograr la mejor alineación posible entre el *grupo de secuencias* en forma simultánea, es decir, componer la mejor *alineación múltiple*. La alineación múltiple de secuencias es uno de los principales problemas en la Biología molecular, ya que es el punto de partida de diversos métodos para el análisis

de la estructura de las proteínas, para interpretar datos del genoma, o como paso intermedio para la construcción de árboles evolutivos de ADN.

La alineación múltiple puede verse como una generalización de los métodos de alineación entre dos secuencias. Infortunadamente, algoritmos rigurosos para encontrar soluciones óptimas han sido computacionalmente caros y tienen aplicabilidad a un número pequeño de secuencias. La complejidad de estos algoritmos crece exponencialmente con el número de secuencias que intervienen, y es impráctico en la mayoría de las aplicaciones reales. De aquí surge la aplicación de métodos heurísticos que dan soluciones aproximadas sacrificando sensibilidad a costa de viabilidad. Una solución en este tipo de alineación es buscar segmentos en cada secuencia que alinee con segmentos en las restantes. Como resultado de este tipo de alineación múltiple es posible encontrar en las secuencias zonas preservadas por la evolución, denominadas *motivos*, los cuales juegan un rol importante en la disposición estructural y funcionalidad biológica de una familia de secuencias. Definiendo como *familia* a un grupo de secuencias procedentes de varias especies que realizan una función similar y tienen un mismo origen evolutivo. En algunas familias los motivos están muy bien definidos y pueden ser descriptos por simples expresiones regulares. Sin embargo, en otros casos los motivos son muy sutiles como para ser descriptos por un patrón preciso y requieren un modelado más complejo: se requiere la determinación simultánea de la *composición* y *ubicación* de los mismos sólo a partir del conjunto de secuencias sin ninguna información adicional. De hecho si se conociese la composición se podría determinar la ubicación y si se conociese la ubicación se podría determinar su composición. En realidad se está frente a un problema NP completo: "el problema de localizar una sub-secuencia común en un grupo de secuencias" [44], [45]. Luego, la formulación de soluciones aproximadas es fundamental. En particular, tales soluciones pueden derivarse a partir de un *modelado estadístico* adecuado sobre las secuencias consideradas. En el problema de detección de motivos, el modelo estadístico corresponde a la distribución de probabilidad conjunta que rige la ubicación y composición del motivo dado el conjunto de secuencias. En otras palabras, el objetivo es calcular *distribuciones de probabilidad a posteriori* sobre variables de interés, el alineamiento.

El *muestreo Gibbs* (Geman *et al*. 1984 [19]) es un método estadístico de características iterativas, que permite la determinación de *distribuciones de probabilidad* y que por lo tanto aplica naturalmente al problema bajo estudio (Lawrence *et al.* 1993 [32], Liu *et al.* 1995 [39]). Tal como se notó anteriormente, la determinación simultanea de ubicación y composición de motivos no es resoluble sólo a partir del conocimiento de las secuencias. Luego, es necesaria la introducción de restricciones a los fines de regularizar el problema. En términos estadísticos, ello puede lograrse mediante la introducción de alguna *distribución de probabilidad a priori* (MacKay, 2003 [42]): el problema se trata bajo el paraguas conceptual del modelo estadístico bayesiano. Desde el punto de vista teórico,

esta tesis considera dos aspectos. En primera instancia se analiza el modelado estadístico bayesiano aplicado a problemas típicos de análisis de secuencias biológicas, como el problema de detección de motivos. Luego, se analiza en forma detallada la aplicación conjunta del método de *muestreo Gibbs* y el modelo bayesiano. Como resultado de este análisis se obtiene un algoritmo computacionalmente viable, cuyo resultado es biológicamente significativo para el problema de detección de motivos. En particular, la complejidad de este algoritmo es lineal con respecto al número de secuencias involucradas.

Desde el punto de vista práctico, en esta tesis se analiza el comportamiento del muestreo Gibbs para el problema de detección de motivos en grupos de secuencias encuadradas en el Proyecto "Caracterización de factores basales de trascripción en parásitos protozoarios", Serra *et al.*[56]. De forma breve, estos grupos se caracterizan por un alto grado de divergencia a nivel de secuencia. Ello deriva en un problema de detección de motivos de características extremas. Para este fin se desarrolló un programa, el *GibbsSM*, que implementa el método.

Esta tesis se organiza como sigue. En el presente capítulo se refieren los conceptos básicos de Bioinformática con relación a la constitución y estructura de las moléculas biológicas. También se introduce el análisis de secuencias a partir de las alineaciones, motivos y perfiles. Por último se presenta el Muestreo Gibbs. En el Capítulo 2 se detallan los métodos que son basamento del Muestreo Gibbs y necesarios para su comprensión. En el Capitulo 3 se explica el método del Muestreo Gibbs para la detección de motivos en un conjunto de secuencias biológicas. En el Capítulo 4, "Parte Experimental", se describe el software GibbsSM y su aplicación a la detección de motivos en secuencias de proteínas relacionadas al Proyecto Serra *et al.* 2003[56]. A los efectos de validación se realiza una comparación de los resultados obtenidos con GibbsSM contra aquellos obtenidos con programas usados tradicionalmente en el ámbito biológico. Se usaron como referencia Gibbs Sampling [32][GIBBS] y MEME [5][MEME] , en el primer caso debido al uso del muestreo Gibbs y en el segundo debido a su amplia difusión. Finalmente se exponen las conclusiones.

La organización propuesta refleja mi tránsito de la Informática pura a la Bioinformática. El objetivo de esta tesis ha sido ampliar, mejorar y afianzar mi formación en un área de investigación de gran actualidad y cambios permanentes. Para lo cual se propone el estudio de un problema real como lo es la localización de motivos. Creemos que este objetivo ha sido cumplido tal como se desprende del trabajo experimental detallado y la inserción lograda en un grupo abierto de investigación en Bioinformática de composición multidisciplinaria.

A riesgo de presentar una simplificación, los temas que son fundamentales como aporte a este estudio, los he desarrollado en detalle.

## 1.2 Conceptos básicos: ADN, ARN y Proteínas

ADN (ácido desoxirribonucleico), ARN (ácido ribonucleico) y proteínas son tres moléculas constituyentes de la vida. Cada una se especializa en realizar una función específica. La información codificada dentro del ADN otorga las características hereditarias del ser tales como el color del cabello, el color de los ojos, así como la predisposición a ciertas enfermedades [41].

El ADN está compuesto por cuatro moléculas: Adenina, Timina, Citosina y Guanina, denominadas en forma general nucleótidos y codificadas con las letras A T C y G. Por otro lado, el ARN tiene una gran variedad de roles, son los transmisores de la información entre el ADN y las proteínas. A su vez, las proteínas son cadenas formadas por 20 aminoácidos diferentes, los cuales se encuentran detallados en el Anexo I. La relación entre la secuencia de nucleótidos y la secuencia de aminoácidos de la proteína es determinada por un mecanismo celular de traducción, conocido de forma general como *código genético* [41]. En el código genético se encuentran las *instrucciones* contenidas en un *gen* - segmento de ADN - que le dicen a la célula cómo hacer una proteína específica.

Las proteínas son las responsables de casi todas las funciones de un ser vivo y de la formación de muchas estructuras vivientes. De hecho, cada proteína tiene un *ordenamiento* o *secuencia de aminoácidos* que es exclusivo en ella y diferente a todas las demás. La secuencia de aminoácidos que caracteriza a una proteína se denomina *estructura primaria*. Esta estructura es muy importante ya que de ella depende la función que desempeña en un organismo. Sin embargo, la complejidad de las moléculas de proteínas no termina aquí. Una vez establecida la cadena de aminoácidos, su estructura primaria, otras interacciones entre los aminoácidos hacen que adopten configuraciones espaciales diversas denominada *estructura secundaria*. Además existen plegamientos sobre sí misma que constituye la *estructura terciaria* que le da un aspecto de ovillo. Varios ovillos a su vez pueden asociarse y formar una única molécula, la *estructura cuaternaria*. A efectos de ejemplificar se observa en la figura 1.1 la secuencia de la estructura primaria de la proteína Metalo-beta-lactamasa de Bacillus cereus y en la figura 1.2 su estructura terciaria.[1]

[1] La estructura se obtuvo de http://www.rcsb.org/pdb/cgi/explore.cgi?pid=178271127825638&pdbId=1BMC
Y el gráfico se realizó con el programa Swiss PDB Viewer 3.7 (http://ca.expasy.org/spdbv/).

>BcII
SQKVEKTVIKNETGTISISQLNKNVWVHTELGSFNGEAVPSNGLVLNTSKGLVLVDSSWDDKLTKELIEMVEKKFQK
RVTDVIITHAHADRIGGIKTLKERGIKAHSTALTAELAKKNGYEEPLGDLQTVTNLKFGNMKVETFYPGKGHTEDNI
VVWLPQYNILVGGCLVKSTSAKDLGNVADAYVNEWSTSIENVLKRYRNINAVVPGHGEVGDKGLLLHTLDLLK

Fig.1.1. Estructura primaria de la proteina Metalo-beta-lactamasa
de Bacillus cereus en formato FASTA

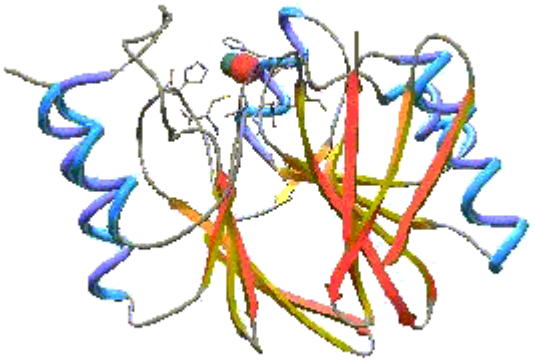

Fig.1.2. Estructura terciaria de la proteina Metalo-beta-lactamasa
de Bacillus cereus

Cada especie de seres vivos es capaz de formar sus propias proteínas y aún dentro de cada especie, cada individuo forma las suyas propias. Esta propiedad de las proteínas se denomina *especificidad*. Las proteínas de un pájaro son distintas de las de un ser humano a pesar de estar constituidas por los mismos aminoácidos. La especificidad de una proteína está relacionada con su estructura primaria. Si bien cada especie de seres vivos posee proteínas especificas existen muchas proteínas muy parecidas entre especies. Dado que el tipo y orden de los aminoácidos que constituyen una proteína determinan cual será la estructura tridimensional final de la misma y considerando que la forma de una proteína determina su función, es fácil deducir que dos proteínas constituidas por los mismos aminoácidos pero dispuestos en distinto orden tendrán distinta función. De igual manera, si a una proteína se le reemplaza un aminoácido por otro de características similares es de esperar que la estructura y por ende la función de dicha proteína no se vea afectada. De aquí se puede deducir que el estudio y la comparación de las estructuras primarias de las moléculas son fundamentales para los investigadores en el ámbito biológico.

Por otra parte, gracias al avance realizado por los distintos proyectos genoma, se encuentra disponible un gran número de secuencias organizadas en distintas Bases de Datos, muchas de ellas públicas. La información contenida en estas Bases de Datos se puede interpretar como *secuencias codificadas* basadas en alfabetos de *d* letras (d=4 para ADN y d=20 para proteínas) sin puntuaciones ni espacios. Empero, esta información se ha escrito en varios formatos dependiendo del origen y nacionalidad de la Base de datos. Estos formatos se refieren a distintas maneras de detallar las características de la estructura primaria de una molécula. El formato FASTA [FASTA] es el más conocido y se emplea en este trabajo por sugerencia del Dr. Serra [56]. En la figura 1.1 se observa la secuencia de la proteína Metalo-beta-lactamasa de Bacillus cereus en este formato.

Inicialmente la Bioinformática trató con la compilación y catalogación de la información molecular obtenida en bases de datos públicamente disponibles como las europeas EMBL [EMBL] y SwissProt [SwissProt] o sus homólogas americanas Protein Identification Resource [PIR] y GenBank [GenBank]. Alrededor de las bases de datos de secuencias han ido apareciendo otras bases de datos especializadas que añaden información deducida a partir de las mismas tales como: bases de motivos como PROSITE [PROSITE] , o compilaciones especializadas como HAEMA [HAEMA] de mutaciones de hemofilia A, o datos complementarios en entornos especializados como IMGT [IMGT] la base de datos de inmunogenética, o de información de interés general como MEDLINE [MEDLINE] de bibliografía médica. Muchas de estas bases de datos están catalogadas y puestas a disposición del público en sitios especializados como el European Bioinformatics Institute [EBI], o el National Center for Biotechnology Information [NCBI] o la European Molecular Biology Network [EMBnet].

Una de las cuestiones científicas más interesantes es como obtener información biológica a partir de estas Bases de datos. Esta tarea es a menudo denominada "data mining"[2]. Sin embargo, "mining" en Base de datos de polímeros es notoriamente diferente al de data mining en otros tipos de bases de datos porque existe una diversidad muy grande de estructuras implementadas. Para tener una idea se puede recorrer la base de datos GenBank de NCBI. Asimismo hay una enorme cantidad de conocimiento biológico y leyes físicas químicas que deben ser aplicadas a esta información para poder interpretarla. En este complejo panorama aparece en escena el problema de cómo extraer, de datos experimentalmente obtenidos, la información suyacente, es decir, cómo de la gran cantidad de secuencias de ADN y proteínas almacenadas en bases de datos, descubrir las propiedades estadísticas o determinísticas que permitan hacer análisis, modelos y juntamente con la generación de estos últimos, obtener hipótesis que se confirmen a través de experimentación. En la sección siguiente veremos algunos métodos que se usan habitualmente en el ámbito biológico para solucionar en parte el problema de buscar en estas bases de datos.

## 1.3 Análisis de secuencias biológicas

A partir del estudio y comparación de las estructuras primarias de las moléculas, los biólogos obtienen pistas para inferir la función de las mismas. Sobre esta base es claro que el rol de la comparación de secuencias es muy significativo. La probabilidad de que una molécula recién secuenciada sea similar a cualquier otra ya secuenciada aumenta con la expansión del tamaño de las bases de datos. En este trabajo se realizan comparaciones de varias moléculas caracterizadas por su estructura primaria, en términos informáticos, el conjunto lineal de *códigos* que caracterizan a las secuencias biológicas.

En algunas áreas de la Biología, la búsqueda de una nueva secuencia de proteína o ADN en una base de datos es una de las tareas más frecuentes. La idea es localizar características relevantes de una secuencia dada comparándola contra todo el cuerpo de conocimiento existente sobre otras secuencias. Se podría comparar dicha secuencia con todas las encontradas hasta el momento, pero esto sería una tarea titánica. Los expertos realizan comparaciones sólo contra secuencias relevantes o subconjuntos bien seleccionados en Bases de datos determinadas, e incluso contra sabiduría concentrada a partir de la base de conocimientos disponibles, obteniendo así resultados más rápidos y certeros. Tales búsquedas producen información de secuencias relacionadas que pueden llevar al descubrimiento de una familia de secuencias que puede ser caracterizada. En este proceso de búsqueda es importante

[2] la traducción al castellano es *Minería de datos*, pero habitualmente se utiliza su acepción en inglés

considerar la *sensibilidad y la selectividad de los algoritmos.* La sensibilidad se refiere a la habilidad de encontrar la mayoría de los miembros relacionados, mientras que la selectividad se refiere a la habilidad de descubrir sólo miembros de la familia en que se está interesado en estudiar. Es importante tener esto presente cuando se interpretan los resultados de las alineaciones y se asigna una función a una secuencia, ya que esta asignación puede darse a través de relaciones transitivas.

Aún hoy en día no se sabe con certeza todos los mecanismos involucrados en la evolución de las secuencias, e incluso lo que se conoce es a veces muy difícil de formalizar. En muchos casos aquello que se puede expresar es complejo como para hacer impracticable cualquier abordaje exhaustivo. Sin embargo, hay situaciones en que interesa una correspondencia exacta, sin mutaciones, ni inserciones ni borrados, sólo coincidencias entre una secuencia dada y cualquiera de la base de datos. Éste es un problema de fácil solución. En otras ocasiones se busca similitud permitiendo mutaciones, inserciones y/o borrados, para contemplar los procesos evolutivos, cuestión no trivial. En este sentido existen metodologías que encuentran similitudes y clasifican las secuencias en orden de calidad acomodando ambigüedad y espacios en la comparación. Algunos métodos resultan demasiado lentos e imprácticos a menos que se usen sobre conjuntos reducidos de secuencias o se disponga de una computadora dedicada. Otros usan soluciones heurísticas, que aunque no garanticen encontrar el mejor ajuste funcionan bastante bien y son mucho más rápidos. En la sección Alineación de secuencias se enuncian algunos métodos conocidos y eficientes que llevan algo más de tiempo pero hallan una respuesta correcta, incluso ordenan los resultados de forma útil para los biólogos.

Finalmente, hay problemas que son imposibles de analizar usando la tecnología existente. Por ejemplo buscar permitiendo traslocaciones, inversiones y otros eventos evolutivos complejos. Estos llevan a un número exponencial de posibilidades que no pueden analizarse prácticamente ni siquiera para dos secuencias.

## 1.4 Alineación de secuencias

La alineación de secuencias biológicas consiste en aparear las mismas de tal manera de lograr que el número de coincidencias sea máximo. En principio, una coincidencia se presenta cuando un residuo[3] de la primer secuencia es igual al de la segunda secuencia.

En el caso de la alineación múltiple, dadas más de 2 secuencias, se busca encontrar *patrones-regiones-motivos en común* en el grupo. Existen muchos métodos y algoritmos que implementan este tipo de alineaciones. Una de estas metodologías es el muestreo Gibbs, objeto de estudio.

### 1.4.1 Alineación con y sin espacios

Un algoritmo trivial es considerar la alineación de 2 secuencias de longitud $n$, donde no se permita la posibilidad de incorporar espacios. Indudablemente existe una única alineación posible.

Pero si se permite incorporar espacios, hay $C_n^{2n} = \frac{(2n)!}{(n!)^2}$ posibles alineaciones totales o globales (Durbin *et al*. 2002).

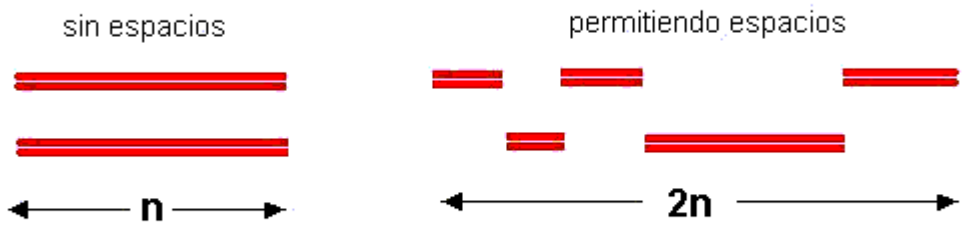


Fig.1.3. Gráfica de la alineación entre 2 secuencias de longitud n.

[3] La palabra residuo hace referencia indistintamente a un aminoácido o a un nucleótido, dependiendo que las secuencias analizadas sean de proteínas o ADN respectivamente.

Es claro que obtener todas las alineaciones posibles a partir de las secuencias no es viable considerando que las longitudes de las mismas superan los cientos de residuos. Sin embargo, se puede pensar en soluciones computacionales. Si se considera a la secuencia como una cadena de símbolos, existen métodos conocidos en Ciencias de la Computación para el manejo de estas situaciones. A continuación veremos algunos.

## Programación Dinámica

Uno de los algoritmos más simples de alineación de 2 secuencias es el que divide el problema en sub-problemas utilizando la idea de la programación dinámica. Éste método es muy usado en el análisis de secuencias, y muchos algoritmos en Bioinformática hacen uso del mismo.

Los algoritmos de programación dinámica usados en este contexto son el de Needleman y Wunsh de 1970 [47], para alineación global, considerada entre las secuencias completas, y el de Smith y Waterman de 1981 [59], para alineación local, entre segmentos de las secuencias. Estos algoritmos garantizan encontrar la alineación óptima, la solución es *exacta*. Sin embargo, en la práctica resultan ineficientes por su orden de complejidad. Sean 2 secuencias una de longitud $n$ y otra de longitud $m$, la complejidad de estos algoritmos tanto en espacio de memoria como en tiempo de orden cuadrático $O(nm) \rightarrow n^2$ [57]. Si se considera que las secuencias de proteínas pueden contener unos cientos de aminoácidos hasta cerca de 5000, y que las moléculas de ADN son de cientos de millones de nucleótidos (que son cortadas en trozos del orden de los 1000 para ser analizadas), y además que este proceso se debe repetir por cada una de las $b$ secuencias que se halla en la Base de Datos (cientos de miles de secuencias), el orden de complejidad pasaría a ser de orden cúbico, $O(nmb)$, y el tiempo de ejecución excesivamente alto [55]. A pesar de este inconveniente, en la actualidad, los métodos de Needleman y Wunsh, y, Smith y Waterman se utilizan como rutinas en muchos programas para alinear segmentos de secuencias que han sido seleccionados por otros métodos, debido a su alineación óptima.

Existen diferentes formas de encarar el problema de la alineación de 2 secuencias [55]:

- ✓ Implementar los algoritmos de programación dinámica directamente en el hardware, que ayudaría a ejecutarse más rápido, sin embargo el costo es muy alto.
- ✓ Usar cientos de procesadores en paralelo y luego integrar los diversos resultados. Pero al igual que el anterior el costo es alto.
- ✓ Usar heurísticas que trabajen mucho más rápido que el algoritmo de programación dinámica, métodos que veremos en la sección siguiente.

A continuación se enuncian los métodos heurísticos más utilizados en Bioinformática.

### 1.4.2 Métodos heurísticos

La *heurística* es una sucesión de normas, introducidas algunas veces con la sola intuición y el espíritu de creación, que permite obtener un resultado que se presume a priori aceptable. La heurística opta por soluciones de compromiso entre tiempo y precisión. Puede ser muy rápida, pero al hacer suposiciones adicionales no suele hallar la alineación *óptima* sino que encuentra una *buena* alineación bajo dichas restricciones. El objetivo de este tipo de metodologías es encontrar una solución aceptable a un problema que, por ciertas circunstancias, no se puede encarar en forma directa.

Las heurísticas más utilizadas para la alineación de 2 secuencias son: FASTA y BLAST

**FASTA**

FASTA [34] es un algoritmo heurístico desarrollado por David Lipman y William Pearson en 1985 y mejorado en 1988. Fue el primer algoritmo ampliamente utilizado para la búsqueda de similitudes en bases de datos. FASTA usa un algoritmo con base heurística para acelerar el proceso de localización de regiones similares. Basa su estrategia en identificar diagonales con mayor número de identidades en una matriz conformada por ambas secuencias dispuestas una en las filas y otra en las columnas. Este proceso no garantiza encontrar la solución óptima. Sin embargo, permite realizar una revisión completa de las dos secuencias en un tiempo proporcional a la suma de los tamaños de las secuencias comparadas, $O(n+m)$.

FASTA hace referencia a un conjunto de programas, cada uno pensado para una combinación específica de un tipo de secuencias y un tipo de base de datos.

**BLAST**

BLAST [1] es otro método heurístico que se usa desde hace muchos años, diseñado por Altschul *et al.* en 1990. BLAST dispone de un mecanismo que da significancia estadística a cada una de las alineaciones utilizando la estadística enunciada por Karlin y Altschul en 1990. El programa proporciona información adicional para permitir al biólogo que se forme una opinión crítica de la validez de los resultados. Al igual que FASTA existe un conjunto de programas BLAST, cada uno pensado para una combinación específica de un tipo de secuencias y un tipo de base de datos.

### 1.4.3 Alineación múltiple

La alineación múltiple de secuencias puede ser vista como una generalización de la alineación de pares de secuencias, donde la complejidad crece exponencialmente con el número de secuencias que intervienen, dando origen a un problema *NP completo* (Durbin *et al.* 2002). La posibilidad de resolver el problema en forma manual queda descartada por la enorme complejidad del problema, limitándose su uso a casos con muy pocas secuencias.

Por analogía con la comparación de pares de secuencias es posible aplicar la misma técnica de programación dinámica al caso multidimensional, pero el crecimiento exponencial de la complejidad del método limitó su aplicación práctica. Por ejemplo, si alinear dos secuencias de 300 residuos tardase un segundo, alinear 3 tardaría 300s y alinear 10 tardaría unos $300^8$s, una cantidad de tiempo mayor que el transcurrido desde el inicio del universo [SHAMIR].

Carrillo y Lipman en 1988 pensaron un método para reducir el volumen del espacio N-dimensional de la matriz de programación dinámica. Este método utiliza alineaciones de parejas de secuencias resultando importantes mejoras en la velocidad. El algoritmo fue implementado por Lipman, Altschul y Kececioglu en el programa MSA (Lipman *et al.* 1989). Sin embargo, la aplicación del mismo se reduce a pequeños grupos de secuencias, MSA puede alinear hasta 10 secuencias de longitudes de 200-300 residuos.

Lo expuesto explica la proliferación de métodos de soluciones aproximadas para conjuntos más grandes de secuencias, e ideas nuevas para implementar algoritmos tales como los clusters, los motivos y los perfiles.

Una de las soluciones propuestas se basa en la formación de *clusters* de secuencias. Para ello, dada una medida del parecido o semejanza entre dos secuencias, se eligen aquel par correspondiente al valor más alto, y se alinean y agrupan entre sí para formar un único grupo o cluster de secuencias. A partir de ese momento este cluster será tratado como una sola secuencia, y el proceso se repite hasta tener un cluster con todas las secuencias que intervenían en la alineación múltiple. Los programas más difundidos con esta metodología son CLUSTALV de Higgins *et al.* 1991 y CLUSTALW en su última versión de Thompson *et al.* 1994.

Otra solución muy utilizada es encontrar segmentos en cada secuencia que presenten características similares del grupo de secuencias. Esta idea llevó a generar programas que, usando métodos estadísticos, *descubren* subsecuencias que caracterizan al grupo, los denominados *motivos* y *perfiles*. En este último grupo se halla el muestreo Gibbs. En la sección siguiente se explicarán estos términos.

Al encarar una alineación múltiple se debe tener en cuenta la elección de las secuencias, ya que los métodos sólo tienen sentido si se supone que se trata de un conjunto de secuencias relacionadas. Cuando esta condición no es satisfecha, se debe estar consciente de que cualquier algoritmo producirá alineaciones no significativas. El biólogo será quien debe suministrar el grupo de secuencias que servirán de entrada a los programas. En esta tesis las secuencias de prueba fueron catalogadas bajo la supervisión del Dr. Esteban Serra del Instituto de Biología Molecular y Celular de Rosario de la Facultad de Ciencias Bioquímicas y Farmacéuticas de la Universidad Nacional de Rosario, República Argentina, en el contexto del Proyecto *Caracterización de factores basales de transcripción en parásitos protozoarios,* CONICET (2003-2004).

## 1.5 Motivos, dominios, patrones, perfiles y sitios

En la evolución biológica, las moléculas que comparten un antepasado común no son precisamente iguales, sin embargo heredan muchas similitudes de la estructura primaria de su antepasado. El análisis de estas similitudes ayuda a la identificación de pequeñas regiones conservadas que pueden identificarse como homólogos remotos, y ciertamente podría dar una idea de la función que una proteína desconocida pudiera tener.

En la figura 1.4. se ilustra un conjunto de 5 secuencias biológicas S= (S1,S2,S3,S4,S5) con longitudes no necesariamente iguales L=( $\ell$ 1, $\ell$ 2, $\ell$ 3, $\ell$ 4, $\ell$5) que presentan *subsecuencias similares* de un mismo ancho *w* que constituyen los *motivos comunes*. La especificación de estos motivos comunes es usualmente descripta a través de *un modelo estadístico*. Esto hace posible pensar en la posibilidad de determinar *modelos estadísticos* que caractericen a las diferentes familias de secuencias.

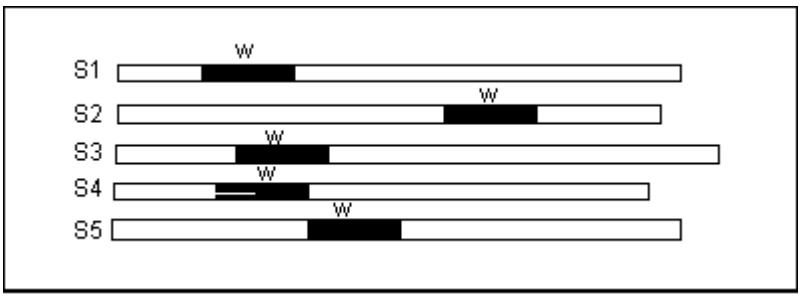


Fig.1.4. Patrones comunes.

Cuando se consulta la literatura referente a estos aspectos, se encuentra que existe cierto descuido y, a veces un mal uso, de algunos términos como *perfiles*, *patrones* y *motivos*; a continuación se explica brevemente cada uno de estos conceptos para poder disipar toda duda al respecto.

## Motivos (*motif*)

Si se observa una alineación múltiple de proteínas homólogas se ve que en algunas columnas el tipo de residuo varían bastante, mientras que en otras los residuos se presentan más conservados. Cuando se observan ciertas columnas cercanas con una alta coincidencia, es decir, cuando se encuentra parte de las secuencias que se conservan más que otras, se está en presencia de *motivos*. Este hallazgo hace pensar que estas zonas podrían caracterizar funcionalmente al grupo de secuencias. En función de esto, la práctica científica ha demostrado que la identificación de motivos es importante en el campo biológico porque se reconocen zonas preservadas por la evolución, las que juegan un rol importante en la estructura y funcionamiento de un grupo de secuencias relacionadas.

El término *motivo* (motif en inglés) hace referencia al objeto biológico (secuencia de residuos), el cual se describe por medio de un patrón (expresión regular). Cuando un motivo es demasiado sutil para poder ser definido con un patrón estándar, se puede utilizar otro tipo de descriptor como el perfil (descripción cuantitativa) o el modelo oculto de Markov (HMM - Hidden Markov Model)[31], los que tienen la intención de reducir exhaustivamente (columna por columna) las propiedades de una familia de proteínas o un dominio. Tanto los perfiles como los HMM hacen posible la identificación de miembros muy distantes de una familia de proteínas al momento de hacer la búsqueda en una base de datos. A partir de la importancia de los perfiles y los HMM en esta tesis, estos conceptos son introducidos en esta sección con una descripción breve, pero su tratamiento es ampliado en el Capítulo 2 en la sección Modelado estadístico de secuencias biológicas.

## Dominio (*domain*)

El *dominio* (domain en inglés) define una unidad estructural independiente en las proteínas. Biológicamente un dominio es una secuencia de aminoácidos que codifica para una estructura secundaria en particular y en general está asociada a una función específica de la molécula. Los dominios usualmente cubren una mayor parte de las secuencias que los motivos. En las bases de datos de dominios como Pfam [PFAM] , un dominio se suele corresponder con aquella zona más similar entre todas las proteínas de una familia, y dentro de él se localizan los motivos.

## Patrones (*pattern*)

Un *patrón* (pattern en inglés) describe un grupo de residuos que constituyen un motivo característico dentro de una secuencia biológica. La sintaxis del patrón permite expresar ambigüedades. Los patrones se representan mediante expresiones regulares del tipo:

AD{DE}X(n)[GP]X{LVIAM}GX(2)G

donde

[abc] = indica que puede haber ambigüedades, cualquiera de los residuos a,b,c
{abc} = indica que los residuos a,b y c NO son aceptados en esa posición
x = para identificar posiciones en las que puede haber cualquier residuo
a(2) = para indicar que ese residuo se debe repetir, 2 residuos a
a(n) = cualquier cantidad de residuos de tipo a

Estas expresiones o patrones se pueden utilizar para caracterizar motivos, indicando cuáles posiciones son más importantes y cuáles pueden variar y qué variaciones pueden sufrir.

## Perfiles (*profile*)

Un *perfil* (profile en inglés) es una matriz de frecuencias de residuos que caracteriza una alineación múltiple de un conjunto de secuencias. Esta matriz tiene como dimensión $d \times w$, donde $d$ es la cantidad de símbolos del alfabeto de las secuencias (d=4 para ADN y d=20 para proteínas) y $w$ es el ancho del alineamiento múltiple. Existen distintos métodos para construir estas matrices. Un método clásico desarrollado por Gribskov en 1987 que requiere una alineación múltiple de las secuencias como dato y hace uso de tablas de comparación de símbolo (Henikoff *et al.* 1992) para convertir distribuciones de frecuencia en pesos. En la figura 1.5 se muestra una ejemplificación de un perfil a partir de las subsecuencias de 5 residuos obtenidas por la alineación de un grupo de 4 secuencias de ADN.

**Alineación**

```
          1 2 3 4 5
          T T C T A
          A T A T A
          A C A T G
          C T G T C
```

**Perfil**

| | Columna1 | Columna2 | Columna3 | Columna4 | Columna5 |
|---|---|---|---|---|---|
| A | 0.50 | - | 0.50 | - | 0.50 |
| T | 0.25 | 0.75 | - | 1 | - |
| C | 0.25 | 0.25 | 0.25 | - | 0.25 |
| G | - | - | 0.25 | - | 0.25 |

Fig.1.5. Ejemplo de un perfil.

## Sitio y no-sitios (site/nonsite)

Se denomina *sitio* a la posición del primer residuo del motivo dentro de la secuencia y *no-sitio* a todas aquellas posiciones de los residuos que no están en el motivo.

## Perfil HMM

Los perfiles basados en modelos ocultos de Markov (Krogh *et al.* 1994) pueden ser usados como una forma más sensible de búsqueda de homólogos remotos y motivos conservados pues están basados en una descripción estadística de la estructura primaria de una familia de secuencias. Un modelo lineal de cadenas ocultas de Markov se corresponde con una secuencia de nodos para cada posición en una alineación múltiple, es decir, modela residuos alineados. HMM usa puntajes para las posiciones específicas de los residuos y otros puntajes para abrir o extender con inserciones o espacios las secuencias (Baldi *et al.* 2001 – Durbin *et al*. 2002) .

La ventaja de este método por encima de otros es que el HMM tiene una base probabilística formal. Los perfiles HMM están basados en métodos que no buscan la exactitud en la alineación sino que describen la familia bajo estudio como un modelo probabilístico.

Existe Bases de datos de perfiles, como la PROSITE [PROSITE] que describe familias de proteínas mediante perfiles simples, o Pfam [Sonnhammer y Eddy., 1997][PFAM] compuesta por perfiles HMMs obtenidos para distintas regiones conservadas en familias de proteínas. El método HMM también es utilizado como herramienta de búsqueda de dichas regiones en las secuencias dadas. Los perfiles HMMs se muestran como el sistema más sensible a la hora de detectar homólogos remotos, de aquí que Pfam sea una de las bases de datos más empleadas en el ámbito biológico.

Existe mucha información respecto a los HMM: Rabiner 1989, Krogh *et al.* 1994, Durbin *et al.* 2002, entre otros. En el Capítulo *Marco teórico* se amplía el estudio de los HMM en el contexto del modelado estadístico de secuencias biológicas.

## 1.6 Muestreo Gibbs

El muestreo Gibbs es un método de inferencia probabilística que permite recuperar información a partir de datos incompletos y/o afectados por ruido. Los problemas con este tipo de datos son más fáciles de encarar suponiendo que los mismos están disponibles. El muestreo Gibbs hace uso de los mecanismos de inferencia bayesiana para hallar la mejor distribución para los datos no observados en un proceso iterativo caracterizado por una gran simplicidad algorítmica.

Los primeros antecedentes del muestreo Gibbs se remontan al algoritmo Metropolis (Metropolis *et al.* 1953, Hasting 1970) en el área de física estadística. Más tarde fue introducido en el contexto de la restauración estadística de imágenes por Geman y Geman en 1984. A partir del trabajo de Lawrence *et al.* en 1993 se comenzó a aplicar el Muestreo Gibbs en el área de Biología molecular para la determinación de alineaciones múltiples de secuencias biológicas. Sea cual fuere la aplicación, la idea fundamental del muestreo Gibbs es la búsqueda de distribuciones de probabilidades a partir de datos completos o incompletos con ruido o sin él. En la literatura tradicional las distribuciones buscadas suelen ser de naturaleza continua, mientras que en Biología molecular lo son de naturaleza discreta.

En el área de Biología molecular se han diseñado una variedad de programas que hacen uso del muestreo Gibbs para la localización de regiones conservadas en secuencias biológicas, tales como el CompareProspector (Yueyi Liu *et al.* 2004) que aplica el muestreo Gibbs a la búsqueda de regiones conservadas a través de las especies con el análisis de elementos de regulación e identificación de eucariotas usando comparación de genomas; el BioProspector (Liu X. *et al.* 2001) que determina las regiones conservadas de motivos en ADN en regiones reguladoras de genes co-expresados; el PGS

(Pseudo-Gibbs Sampler, Stephens *et al*. 2001) que permite construir haplotipos parciales; el AlignACE (Aligns Nucleic Acid Conserved Elements) (Roth *et al.* , 1998) que encuentra elementos conservados en un sistema de secuencias de ADN; o el GMS (Liu *et al.* , 1995; Neuwald *et al.* , 1993) que permite buscar múltiples motivos en secuencias de proteínas. Además existen proyectos basados en el estudio e investigación de métodos aplicando el muestreo Gibbs como el BUGS (Bayesian inference Using Gibbs Sampling)[BUGS], 1989-2004, en el MRC Biostatistics Unit, Cambridge, UK.

En el capítulo siguiente se plantean los métodos estadísticos que son base del muestreo Gibbs. Se focaliza el estudio sobre la estadística bayesiana, los métodos de Monte Carlo, las cadenas de Markov, y la metodología de Markov Chain Monte Carlo, para luego explicar en detalle y con fundamento teórico los aspectos principales del muestreo Gibbs que es objeto de estudio en esta tesis. En el Capítulo 3 se enfoca el muestreo Gibbs a la detección de motivos en secuencias biológicas, para luego plantear la experimentación y comprobación del método en el Capítulo 4.

# Capítulo 2: Marco teórico

*"Lo esencial es invisible a los ojos"*
*Antoine de Saint Exupery*

## 2.1 Modelado estadístico

La Estadística proporciona elementos teóricos y prácticos que nos ayudan a estudiar la realidad. Dentro de este contexto, hace uso de marcos conceptuales que permiten organizar los esfuerzos para comprender el mundo real. Estos marcos son los *modelos*. Un modelo proporciona una representación simplificada que conserva la esencia de una realidad, facilitando el estudio de las interrelaciones entre variables dependientes y/o independientes del sistema bajo estudio. A un modelo no se le exige que sea 'verdadero', sino que sea 'útil', de acuerdo a los objetivos para los cuales fue creado. Es claro que no debe confundirse con la realidad que intenta representar; el modelo es una creación que busca ayudar a comprender una realidad bajo ciertas condiciones y no es la realidad misma.

El modelado y análisis estadístico, incluyendo el conjunto de datos, la construcción del modelo probabilístico, la incorporación de opiniones de expertos, la interpretación del modelo y de los resultados, y la predicción a partir de los datos, forman una parte esencial del método científico en diversos campos. La clave del modelado estadístico es hacer *inferencias* sobre los datos observados para obtener ciertas conclusiones sobre la realidad del sistema bajo estudio.

En Bioinformática uno se encuentra constantemente con el problema de inferir modelos. Sin embargo, la información que se dispone está compuesta de datos observables y otros que no lo son. Esto constituye un problema a la hora de modelar la realidad. Un recurso para solucionar este tipo de problemas es el modelado estadístico bayesiano. Aunque las bases de la estadística bayesiana datan de hace más de 2 siglos (Bayes, 1763), no es hasta la última década cuando empieza a verse un uso creciente de este enfoque en diversos campos de la Ciencia. Una de las razones que explica esta realidad y que a la vez anuncian un impetuoso desarrollo futuro es la absoluta necesidad de cálculo computarizado para la resolución de algunos problemas de cierta complejidad.

En particular, el modelado bayesiano es directamente aplicable al problema de localización de motivos en un conjunto de secuencias biológicas. El objetivo en este caso es la determinación de los parámetros de distribución de probabilidad que rige la ubicación y composición de los motivos *dadas* las secuencias o datos observables. Estos dos objetivos, ubicación y composición, no son resolubles sólo a partir del conocimiento de las secuencias: la composición determina la ubicación y la ubicación determina la composición. De hecho, se necesita un conocimiento a priori que permita regularizar el problema. El modelado estadístico bayesiano es capaz de introducir de forma muy natural toda información a priori y/o a posteriori expresables bajo alguna función de distribución de probabilidad. La solución a este tipo de problemas se encuadra dentro de los denominados *Bayesian data-missing problem* y son a la vez los típicos problemas de la Bioinformática, difíciles pero interesantes. En la sección siguiente se aborda el modelado bayesiano con vista a tratar los problemas con datos ocultos.

## 2.2 Modelado bayesiano

El modelado bayesiano es un enfoque alternativo para el análisis estadístico de datos que, en buena medida, se contrapone a los métodos que proceden de lo que se ha denominado "estadística frecuentista". El interés por el teorema de Bayes trasciende cuando se amplía a otro contexto en el que la probabilidad no se entiende exclusivamente como la frecuencia relativa de un suceso a largo plazo, sino como el grado de convicción personal acerca de que el suceso ocurra o pueda ocurrir (definición subjetiva de la probabilidad). Al admitir un manejo subjetivo de la probabilidad, el análisis bayesiano nos permite emitir juicios de probabilidad sobre una hipótesis y expresar por esa vía un grado de convicción al respecto, tanto antes como después de haber observado los datos. Esta manera de razonar de la inferencia bayesiana, radicalmente diferente a la inferencia clásica o frecuentista (que desdeña en lo formal toda información previa de la realidad que examina), es sin embargo muy cercana al modo de proceder cotidiano e inductivo de los científicos. Debe subrayarse que esta metodología, a diferencia del enfoque frecuentista, no tiene como finalidad producir una conclusión dicotómica (significación o no-significación, rechazo o aceptación) sino que cualquier información empírica, combinada con el conocimiento que ya se tenga del problema que se estudia, *actualiza* dicho conocimiento y la trascendencia de dicha visión actualizada no depende de una regla mecánica.

El método bayesiano para el análisis de datos se centra en el cálculo de las distribuciones de probabilidad condicionales sobre las variables de interés dados los datos observados. Estas variables de interés pueden ser pensadas como observaciones *futuras* no observables al inicio del problema.

El análisis bayesiano comienza normalmente con un modelo de probabilidad completo: la distribución de probabilidad conjunta sobre todas las variables bajo estudio. Luego la aplicación del teorema de Bayes permite calcular las distribuciones de probabilidad condicional sobre las variables de interés. Estas distribuciones son denominadas *distribuciones a posteriori*. En su forma más simple, si denotamos por $q$ a la variable de interés y por $D$ los datos, el teorema de Bayes afirma que:

$$P(q \mid D) = \frac{P(D \mid q)\, P(q)}{P(D)} \qquad (2.1)$$

En (2.1) $P(q)$ es considerada como una afirmación probabilística del conocimiento previo sobre $q$, denominada probabilidad *a priori.* De modo similar, $P(q \mid D)$ se vuelve una afirmación probabilística reconsiderada de nuestro conocimiento sobre $A$ a la luz de los datos, denominada probabilidad *a posteriori.* Mientras que $P(D \mid q)$ es la probabilidad de observar el conjunto de datos $D$ en un universo donde se verifica $q$. Finalmente, $P(D)$ representa la probabilidad de observar el conjunto de datos D, y a todos los efectos sólo juega como una constante de normalización. Cierto es que en el análisis bayesiano toda información es asociada a una distribución de probabilidad. Por lo tanto, puede decirse que este teorema establece las bases del aprendizaje estadístico en el contexto probabilístico.

### 2.2.1 Selección del modelo

Una tarea central en el ambiente científico es desarrollar y comparar modelos que expliquen lo mejor posible los datos dados. Empero, la comparación de modelos es una tarea difícil porque no es posible elegir sencillamente un modelo que ajuste de la mejor manera a los datos. En esta tarea de modelaje están involucrados dos niveles de inferencias. En un primer nivel de inferencia, se supone un modelo $M$ como válido (la hipótesis). El modelo lleva implícito un número de variables $q$ que deben ajustarse a los datos $D$. En el primer nivel, se debe inferir qué valores de las variables de interés $q$ del modelo $M$ se pueden dar en el contexto de los datos $D$. Las variables $q$ pueden ser pensadas como parámetros libres del modelo. Visto desde este punto, ajustar el modelo $M$ a los datos $D$ implica inferir que valores $q_i$ tomarán cada uno de esos parámetros dado los datos. Frecuentemente, los resultados de esta inferencia son a menudo sintetizados por los valores más probables de los parámetros $q$.

El segundo nivel de inferencia se refiere a la tarea de comparación de modelos. Aquí, se deben comparar los modelos a la luz de los datos, y asignar un tipo de prioridades para cada uno de ellos. En esta tesis sólo se refieren de manera formal al primer nivel de inferencia. Una metodología formal para el segundo nivel de inferencia puede encontrarse en MacKay , 1992.

Cada modelo *M* de parámetros $q$ es definido por su forma funcional y por 2 distribuciones de probabilidades: la distribución a priori $P(\theta)$ la cual define qué valores pueden tomar los parámetros $q$ del modelo; y las predicciones $P(D|\theta)$ que el modelo hace sobre los datos $D$ cuando estos parámetros tienen un valor particular $q^*$. Es de notar que modelos con la misma parametrización pero diferente distribución a priori son modelos diferentes. De esta forma, el modelado bayesiano puede ser descripto en 3 pasos (Gelman *et al.* , 1995):

(a) *Establecer* un modelo $M$ a través de una distribución de probabilidad conjunta $P(D, \theta)$ que capture las relaciones entre las variables del problema: datos $D$ y parámetros libres $\theta$.
(b) *Estimar* los parámetros libres $\theta$. Este paso implica el cálculo de la distribución de probabilidad a posteriori $P(q|D)$ de los parámetros $\theta$ dados los datos $D$, y el uso de métodos de gradiente para la determinación del valor más probable de los parámetros. Este tipo de estimación se conoce generalmente como estimación MAP o *Maximum a Posteriori*.
(c) *Ajustar* el modelo $M$ con los valores hallados en (b) con posible repetición de los pasos (a) y (b). Este paso implica un proceso de comparación de modelos, muchas veces informal y empírica, hasta la obtención del modelo final (MacKay , 1992).

Usualmente el paso (a) se resuelve estableciendo una relación probabilística entre los datos observados $D$ y los parámetros libres $\theta$. De forma estándar ello implica el cálculo de la función de likelihood $P(D|q)$. El paso siguiente es la elección de una distribución a priori $P(q)$. Tal distribución debe ser matemáticamente manejable y a la vez significativa para el problema bajo estudio. Finalmente, la distribución de probabilidad conjunta puede representarse como el producto entre la función de likelihood y la distribución a priori, esto es $P(D,q) = P(D|q)\,P(q)$.

En el paso (b) se resuelve planteando el teorema de Bayes (2.1). Sin embargo, es de notar que $P(D)$ puede considerarse como una constante de normalización, por lo cual la distribución de *probabilidad a posteriori* resulta proporcional al producto del *likelihood* y la *probabilidad a priori.*

$$P(\theta \mid D) \propto P(D \mid \theta)\, P(\theta) \qquad (2.2)$$

$$posteriori \propto likelihood \times priori$$

De este modo se obtienen ciertos valores de $\theta$ que satisfacen el modelo elegido en (a). Así se obtiene un conjunto de modelos $M_i = \{\theta\}_i$ posibles sobre un mismo conjunto de datos $D$.

En el paso (c) se parte del conjunto de parámetros $\theta$ hallados en (b) y se trata de ajustar el modelo. El objetivo es encontrar el *mejor* modelo dado los datos y la distribución a priori del modelo, es decir, el modelo más probable que se ajuste mejor a los datos dados. Esto significa buscar los $\theta$ que den la mayor probabilidad a posteriori dado $D$ (Bernardo *et al.* 1994). Lo cual significa encontrar un conjunto de parámetros $\theta$ que maximice $P(\theta \mid D)$. Este proceso se denomina estimación del *máximo a posteriori (MAP)*

$$\theta_{MAP} \equiv \max_{\theta} P(\theta \mid D) = \max_{\theta} \frac{P(D \mid \theta) P(\theta)}{P(D)} \qquad (2.3)$$

En la ecuación(2.3) , $P(D)$ no depende de los parámetros $\theta$ y por lo tanto es irrelevante en la optimización.

$$\theta_{MAP} \equiv \max_{\theta} P(\theta \mid D) = \max_{\theta} P(D \mid \theta)\, P(\theta) \qquad (2.4)$$

Si se considera que las probabilidades $P(\theta)$ son igualmente probables, resulta

$$\theta_{MLE} \equiv \max_{\theta} P(D \mid \theta) \qquad (2.5)$$

$\theta_{MLE}$ se denomina estimador de *máxima verosimilitud* o *máximo likelihood (MLE).*

Como se observa en (2.5) el objetivo de la *estimación del máximo likelihood* (MLE) es determinar valores de los parámetros que mejor expliquen los datos observados sin suponer distribuciones previas de probabilidad. A diferencia del MLE (2.5), en el proceso MAP (2.3) se asume un conjunto de modelos posibles parametrizados por $\theta$ con peso $P(\theta)$.

A pesar que la metodología del cálculo del MAP o del MLE es directa, su implementación en general es complicada. Esto es debido a que en la gran mayoría de los problemas de interés práctico, los datos pueden ser incompletos. La incomplenitud puede deberse a la simple pérdida debida a algún proceso de discretización o, en forma más general, a la necesidad de contar con información adecuada bajo la forma de muestreos de variables de interés pero imposibles de observar. Ambos casos pueden tratarse bajo un único contexto: *los problemas bayesianos con datos ocultos* (Bayesian data-missing problem*)*. Una cuestión importante en este punto es cómo se podría realizar computacionalmente estos cálculos. Los estadísticos han desarrollado varios algoritmos, entre ellos, el clásico algoritmo de Expectation-Maximization para MLE (Dempster *et al.* 1977) y el muestreo Gibbs en general (Gelfand *et al.* 1990). En las secciones siguientes se analizará como encarar los problemas con datos no observables y estos dos algoritmos que vienen en ayuda de su solución.

### 2.2.2 Problema bayesiano con datos ocultos

Una vez que un modelo $M(q)$ es elegido, los parámetros $q$ del mismo deben ser inferidos a partir de los datos. Si *x* es una observación de un proceso aleatorio, por ejemplo un muestreo, cuya densidad de probabilidades $f(x \mid M(q))$ tiene una forma particular (por ej. gaussiana, multinomial, exponencial, Dirichlet, etc.), el objetivo es determinar los parámetros $q$ *que ajusten* dicha función a los datos observados. De ahora en más, con el objetivo de simplificar la notación, se referirá con $f(x \mid q)$ a $f(x \mid M(q))$.

Sea $A$ la variable de interés y sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto de observaciones independientes e idénticamente distribuidas sobre una función de densidad $f(X \mid q)$ que depende de $A$. De hecho en un problema bayesiano típico se considera una distribución a priori $f(q)$, y se plantea simplemente la inferencia sobre la distribución a posteriori de $q$ calculada por el teorema de Bayes:

$$f(q \mid X) = \prod_{i=1}^{n} f(x_i \mid q)\, f(q) / f(X) \qquad (2.6)$$

Sin embargo, en muchas situaciones reales ocurre que $X$ no es completamente observable. Una parte de los datos son observados, $X_{obs}$, y otros no lo son, $X_{mis}$. Como hipótesis se supone que los datos *ocultos* o *missing* son completamente aleatorios. ¿Cómo solucionar entonces este problema?

Sea $X = (X_{obs}, X_{mis})$ el conjunto de los datos observados y no observados. Para estimar $q$ por el método de MLE se debe evaluar la función de likelihood sobre los datos observados, los cuales pueden ser derivados por marginalización de los datos ocultos sobre la función de likelihood sobre los datos completos. Este cálculo involucra la integración sobre los datos no observables:

$$f(q \mid X_{obs}) = f(X_{obs} \mid q) \equiv \int f(X_{obs}, x_{mis} \mid q) dx_{mis} \qquad (2.7)$$

A menudo esta integral es muy difícil de evaluar analíticamente. Se verá como puede resolverse este problema usando el método bayesiano para el problema de datos no observados (Liu 1994). Basado en una fórmula simple se puede escribir la probabilidad a posteriori como:

$$f(q \mid X_{obs}) = \int f(q \mid X_{obs}, x_{mis}) f(x_{mis} \mid X_{obs}) dx_{mis} \qquad (2.8)$$

La idea de la múltiple imputación puede ser aplicada a estos casos de incomplenitud. Esto es obtener múltiples valores $X_{mis}^{(1)}, ...., X_{mis}^{(m)}$ como muestreo aleatorio de la distribución $f(X_{mis} \mid X_{obs})$, conformando un conjunto completo de *m* datos *X*. Con este set de datos completos *X* y aplicando el teorema de grandes números, se puede aproximar la distribución a posteriori de $q$ por el valor medio de las distribuciones posteriores del conjunto de datos completo:

$$f(q \mid X_{obs}) \simeq \frac{1}{m}\left\{ f(q \mid X_{obs}, X_{mis}^{(1)}) + ..... + f(q \mid X_{obs}, X_{mis}^{(m)}) \right\} \quad (2.9)$$

Sin embargo, una dificultad para realizar una imputación *exacta*, es que en muchos problemas prácticos es imposible muestrear directamente un $X_{mis}$ de una distribución $f(\bullet \mid X_{obs})$. Esta dificultad puede ser salvada a través de los métodos denominados Markov Chain Monte Carlo (MCMC), los que permiten resolver de forma simple los problemas bayesianos con datos ocultos. Asimismo (2.9) puede interpretarse como una aproximación de Monte Carlo para $h(x) = f(q \mid X_{obs}, X_{mis}) f(X_{mis} \mid X_{obs})$. En la sección siguiente se analizará en forma breve tanto la aproximación de Monte Carlo como los MCMC.

## 2.3 Método de Monte Carlo

El método de Monte Carlo fue desarrollado para calcular integrales complejas a partir de generación de números aleatorios. Se plantea el cálculo de la integral

$$\int_a^b h(x)dx \qquad (2.10)$$

donde *h(x)* es una función compleja de integrar. Si se puede descomponer *h(x)* en una función *f(x)* más simple y una función de densidad de probabilidad *p(x)* en el intervalo *(a,b)*, la integral se podría escribir de la siguiente manera

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = E_{p(x)}[f(x)] \qquad (2.11)$$

Notar que si se considera a $f(x) = f(q \mid X_{obs}, X_{mis})$ y a $p(x) = f(X_{mis} \mid X_{obs})$ se está en presencia de la ecuación (2.8).

De esta forma la integral puede ser expresada como la esperanza de *f(x)* sobre la densidad de probabilidad *p(x)*. Entonces el método de Monte Carlo afirma que se puede encontrar una cantidad considerable de variables aleatorias $x_1, ...., x_n$ que siguen la densidad de probabilidad *p(x)*, y que la integral se puede aproximar a la media de la función *f(x)* evaluada en dichas variables:

$$\int_a^b h(x)dx = E_{p(x)}[f(x)] \simeq \frac{1}{n}\sum_{i=1}^{n} f(x_i) \quad \textit{integración de Monte Carlo} \quad (2.12)$$

De esta forma la esperanza de una función $f$ de una variable aleatoria *x* puede ser aproximada por $\frac{1}{n}\sum_{i=1}^{n} f(x_i)$ donde los $x_i$ siguen una densidad de probabilidad *p(x)* y tiende a $E[f(x)]$ cuando $n \to \infty$.

Del mismo modo, la estimación Monte Carlo de una función de *t* variables $(q^{(1)},....,q^{(t)})$ está dada por $E\,[\,f(q^{(1)},....,q^{(t)})]_n \simeq \frac{1}{n}\sum_{i=1}^{n} f(q_i^{(1)},....,q_i^{(t)})$ y tiende a $E[\,f(q^{(1)},....,q^{(t)})]$ cuando $n \to \infty$.

La integración de Monte Carlo también puede ser usada para aproximar la distribución a *posteriori* requerida en el análisis bayesiano. Si se considera la integral $J(y) = \int f(y\,|x)p(x)\,dx$ se la puede aproximar por $J(y) \simeq \frac{1}{n}\sum_{i=1}^{n} f(y\,|x_i)$ donde las $x_i$ siguen la densidad *p(x).*

Naturalmente en todos los casos para que la sumatoria sea sencilla de calcular se debe elegir *p(x)* de manera que sea fácil de muestrear. La precisión de este valor depende tanto del tamaño de la muestra *n* como de la distribución de muestreo *p(x).* Para aproximar una integral existe una infinita cantidad de estimadores. Un aspecto importante del método de Monte Carlo se refiere al diseño de técnicas de reducción de varianza para dichos estimadores. Una de las técnicas más sencillas consiste en elegir una distribución de muestreo adecuada. Generalmente se requiere una *p(x)* que sea fácil de simular y tenga una forma similar a la *f(x)* , (MacKay 2003).

Finalmente, los esquemas de muestreo a menudo son simples si se dispone de la distribución apropiada *p(x)*. Aún cuando *p(x)* no sea sencilla se puede resolver utilizando el muestreo por importancia (importance sampling) o mediante la aplicación del algoritmo de Metropolis-Hastings (Metropolis *et al.* 1953, Hasting 1970).

El muestreo por importancia es un proceso no iterativo cuya estrategia es encontrar una distribución simple *p(x)* diferente pero cercana a *q(x)* con la cual se generen los muestreos, resultando:

$$\int_a^b f(x)q(x)dx = \int_a^b f(x)\frac{q(x)}{p(x)}\,p(x)dx = E_{p(x)}[\,f(x)\frac{q(x)}{p(x)}]$$

Esto forma la base del método. Si se generan *n* muestreos sobre *p(x)*, y aplicando (2.12) a esta situación se obtiene:

$$\int_a^b f(x)q(x)dx \simeq \frac{1}{n}\sum_{i=1}^{n} f(x_i)(\frac{q(x_i)}{p(x_i)})$$

Si estos puntos $x_i$ son muestras sobre *p(x)* entonces se puede hallar un estimador para la integral. Pero cuando se generan los muestreos, puede suceder que se encuentren $x_i$ donde el valor de $q(x_i)$ sea mayor que el de $p(x_i)$, el estimador se sobrevalúa; o casos inversos, donde $q(x_i)$ sea menor que $p(x_i)$, en este caso el estimador se subvalúa. Para contemplar estas situaciones, el método de muestreo por importancia evalúa el error de la distribución introduciendo *pesos* $w_i = \frac{q(x_i)}{p(x_i)}$. Estos pesos se usan para ajustar la *importancia* de cada punto en el estimador.

Sin embargo, el muestreo por importancia funciona bien cuando la densidad propuesta *p(x)* es similar a *q(x)*. De hecho, en problemas complejos es dificultoso crear una única densidad *p(x)* que tenga esta propiedad. En esta situación se hace uso del segundo método, el Metrópolis-Hasting.

En contraste con el método de muestreo por importancia, donde las $\{x_i\}$ son muestreos independientes de una distribución propuesta, el método Metrópolis-Hasting es un proceso iterativo que produce muestreos correlacionados. Cada muestreo $x_i$ toma una distribución de probabilidad que depende del valor previo $x_{i-1}$, generando así una cadena de Markov de muestreos. Esta cadena tiene que ser ejecutada un tiempo considerable para lograr generar muestreos independientes sobre *p(x)*. Sin embargo, el problema persiste si no se conoce dicha distribución, como es el caso de la incomplenitud. Ciertamente, la aproximación de Monte Carlo sólo es aplicable a la inferencia bayesiana de modelos con datos completamente observables. Una solución a los problemas con datos incompletos es generar el muestreo utilizando cadenas de Markov que tengan como distribución estacionaria la *p(x)* buscada, es decir, aplicar Markov Chain Monte Carlo (MCMC). El método de Metropolis-Hasting es un claro ejemplo de un MCMC. Se verá a continuación una breve reseña de cadenas de Markov para luego introducir los métodos de MCMC.

## 2.4 Cadenas de Markov

Una cadena de Markov es un tipo de proceso estocástico de gran importancia en Bioinformática. Suelen describir procesos discretos que evolucionan en el tiempo (generaciones) o en el espacio (secuencias biológicas).

Sea $X_t$ el valor de una variable aleatoria en un tiempo *t*, y sea un conjunto finito de valores posibles de *X*, los posibles estados del sistema $\{s_i, i \geq 0\}$. Se dice que esta variable aleatoria es un *proceso de*

*Markov* si la probabilidad de transición entre diferentes valores en el espacio de estados sólo depende del estado actual de la misma:

$$\Pr(X_{t+1} = s_j \mid X_0 = s_k, \ldots, X_t = s_i) = \Pr(X_{t+1} = s_i \mid X_t = s_i)$$

Una *cadena de Markov* es una secuencia de variables aleatorias $(X_0, X_1, \ldots, X_n)$ generadas por un proceso de Markov. La característica principal de una cadena de Markov es su *falta de memoria*: tan solo importa el estado actual para predecir el estado futuro. Precisamente $X_{t+1}$ sólo depende de $X_t$ y no de los anteriores $X_0, X_1, \ldots, X_{t-1}$. De esta forma, la probabilidad de transición del estado *(t+1)* depende sólo del estado anterior *(t)*.

El estado inicial de una cadena de Markov suele ser aleatorio y en general se considera que su valor viene determinado por una distribución de probabilidad inicial. Este valor suele ser *olvidado* después de varias iteraciones.

Bajo ciertas condiciones de ergodicidad, que en la mayoría de los casos prácticos se cumplen, una cadena de Markov converge a una distribución *estacionaria.* La convergencia se refiere a que la distribución de probabilidades toma un valor en el límite, y la condición estacionaria significa que en ese límite se llega a un invariante. Una distribución $p(x)$ es un invariante de la probabilidad de transición de un estado $x^{(t)}$ a otro $x$, es decir, $P(x \mid x^{(t)})$ si $p(x^{(t)}) = \int P(x \mid x^{(t)}) p(x)\, dx$. Además, si la cadena de estados es ergódica esta distribución estacionaria es única. Una cadena se dice ergódica si es irreductible y aperiódica. Irreductible significa que cualquier conjunto de estados puede ser alcanzados por otros estados en un número finito de movimientos. Por último, un estado $i$ es periódico con periodo $k>1$ si $k$ es el menor número tal que todas las trayectorias que parten del estado $i$ y regresa al estado $i$ tienen una longitud múltiplo de $k$. De aquí que una cadena es aperiódica si sus estados no son periódicos.

Una complicación de este método es determinar cuánto tiempo debe pasar para que la cadena de Markov llegue a un estado de equilibrio. Este problema ha llevado a investigaciones matemáticas de los límites de convergencia en diferentes tipos de cadenas de Markov. Desarrollos más recientes incluyen el uso del MCMC, el cual es aplicable tanto a la inferencia bayesiana de modelos con datos completos como con datos incompletos. La forma más simple de esta última metodología es el muestreo Gibbs, tema central de esta tesis.

En la sección siguiente se describe la metodología MCMC teniendo como meta a la explicación conceptual del muestreo Gibbs.

## 2.5 Markov chain Monte Carlo (MCMC)

La aproximación de Monte Carlo es una herramienta útil aunque para modelos complejos hallar una distribución de muestreo razonable puede ser una tarea extremadamente difícil.

El objetivo del muestreo es generar un conjunto de variables que sigan una determinada función de probabilidad $p(x)$. En vez de crear muestreos independientes, los métodos de MCMC construyen una cadena Markov con un muestreo de variables dependientes $x_1,....,x_n$ que tienen una distribución estacionaria $p(x)$.

Sea el caso estudiado en la sección 2.2 donde se dispone de un modelo con datos $D$ y variables de interés $A$, y se quiere muestrear sobre la distribución a posteriori $f(A|D)$. El método MCMC genera una cadena de Markov con muestreos $A^{(0)},...., A^{(m)}$ que tienen como distribución estacionaria la $f(A|D)$. El método MCMC comienza suponiendo un muestreo inicial $A^{(0)}$ de $f(A|D)$ y propone una densidad de probabilidad $q(A)$ para los muestreos. Esta densidad $q(A)$ debe ser tal que $q(A) = f(A)/K$, donde $K$ es una constante de normalización desconocida y difícil de calcular (Walsh 2002). En cada iteración se realiza un muestreo sobre $q(A)$. En la iteración *(t+1)* el muestreo sobre *q* está dado por $y \sim q(A|A^{(t)})$. Luego del muestreo se calcula la relación de densidad de probabilidad entre este muestreo y el valor actual $A^{(t)}$, donde el valor de *K* se cancela:

$$a = \frac{q(y|D)}{q(A^{(t)}|D)} = \frac{f(y|D)}{f(A^{(t)}|D)}$$

Si $a > 1$ significa que la densidad se ha incrementado respecto al estado anterior, y se acepta este muestreo como un nuevo $A^{(t+1)}$. En caso contrario se descarta y se vuelve a muestrear. De este modo se obtiene una secuencia $A^{(0)},...., A^{(m)}$ de estados. Si la cadena resulta ergódica y luego de una considerable cantidad de iteraciones (período burn-in), converge hacia una distribución de equilibrio que se *acerca* a la $f(A|D)$ buscada.

Los métodos MCMC tienen dos ventajas que los hacen útiles para el análisis bayesiano. Primero, se puede elegir *q* que sea fácil de simular. Una buena elección de *q* (que puede depender de los datos) simplifica el algoritmo. Si se elige una *q* que propone valores que son muy cercanos a $A^{(t)}$ la cadena se moverá muy lentamente y tomará un largo tiempo para converger a la distribución estacionaria. Si *q* propone nuevas variables que son lejanas a $A^{(t)}$, las propuestas casi siempre serán rechazadas y otra vez la cadena convergerá lentamente. La elección de *q* será un proceso de prueba y error. Esencialmente, la única restricción en la elección de *q* es que ésta resulte en una cadena irreductible y aperiódica. La segunda ventaja es que no hay necesidad de calcular la constante de normalización *K* porque se cancela en la ecuación del $a$ .

Si se aplica esta metodología al caso de una distribución de probabilidad *p(x)* no sencilla, como se vio en Monte Carlo, se debe generar una cadena de Markov aperiódica e irreductible con distribución de probabilidad estacionaria *p(x)*. De esta forma se calcula una $\overline{f}_n = \frac{1}{n}\sum_{t=1}^{n} f(x^{(t)}) \to E_{p(x)}[f(x)]$ con $n \to \infty$, donde $\overline{f}_n$ es la media ergódica, que precisamente resuelve la integración de Monte Carlo (2.12) para distribuciones *p(x)* complejas.

En particular, el uso del muestreo iterativo MCMC para el problema bayesiano con datos ocultos fue tratado por Tanner y Wong (1987), Li (1988), y Gelfand y Smith (1990) ilustraron su conexión con el muestreo Gibbs. Sobre la hipótesis que con los datos completos $X = (X_{obs}, X_{mis})$ la distribución a posteriori de $A$ es sencilla de calcular, la idea es iterar entre los *missing data* y los *parámetros.* De esta forma, el proceso iterativo se realiza en 2 pasos: muestrear $A \sim f(A \mid X_{obs}, X_{mis})$ y luego muestrear $X_{mis} \sim f(X_{mis} \mid X_{obs}, A)$. Finalmente, se va construyendo una cadena de Markov con distribución de equilibrio $f(A \mid X)$. Solucionando así el problema inicial planteado en la sección de Modelo bayesiano con datos ocultos representado por la ecuación (2.9).

Sin embargo, una cuestión no resuelta en MCMC es cuan larga debe ser la cadena para que llegue a un estado estacionario. Geyer (1992) propone una cadena de longitud amplia, en cambio, Gelman *et. al.*(1992) proponen múltiples cadenas cada una de las cuales comienzan con diferentes valores iniciales. En la práctica, típicamente los primeros 1000 elementos de la cadena de Markov son descartados y luego se usa un test de convergencia (Cowles *et al.* 1996) para calcular si el estado estacionario es efectivamente alcanzado.

Finalmente, el muestreo Gibbs es un caso especial del algoritmo de MCMC: Si $A$ es un parámetro multidimensional, $A=\{a_1, a_2, ..., a_n\}$, el muestreo Gibbs actualiza secuencialmente cada componente de $A$ desde la distribución condicional completa *fijando* los valores de todas las otras componentes y de los datos:

$$P(a_k \mid a_1, ....., a_{k-1}, a_{k+1}, .....a_n, D)$$

Para muchos modelos usados en la práctica común, incluso aquellos que siguen una distribución a posteriori compleja, muestrear desde este tipo de distribución a posteriori condicional es a menudo una tarea relativamente simple. En la sección siguiente se analiza el muestreo Gibbs el cual hace uso de la metodología de resolución recién enunciada.

## 2.6 Muestreo Gibbs

El muestreo Gibbs es un caso especial de los algoritmos MCMC que fue propuesto por Geman y Geman en 1984 para el procesamiento de imágenes. El método genera indirectamente variables aleatorias desde una distribución marginal $p(x)$ sin necesidad de calcular la densidad conjunta. El muestreo goza de una formulación algorítmica muy simple, pero esto no debe confundirse con trivialidad conceptual. Se espera que como resultado del proceso iterativo se obtenga una distribución límite que aproxime a la distribución de los datos observados. El proceso límite es en realidad la distribución de probabilidad estacionaria de un proceso Markov construido por el propio proceso de actualización de parámetros. Suponiendo que las condiciones de ergodicidad se cumplen, lo cual se da con frecuencia para datos reales, el método Gibbs permite muestrear distribuciones complejas. En la sección siguiente se formaliza este concepto matemáticamente.

### 2.6.1 Explicación en términos matemáticos

La idea básica del muestreo Gibbs es construir una cadena de Markov donde la distribución de equilibrio sea $p(x)$. La esencia del método radica en considerar una distribución condicional univariante, es decir, la distribución cuando *todas las variables son fijas menos una.* Tales distribuciones condicionales son mucho más fáciles de simular que las complejas distribuciones conjuntas y, usualmente, tienen formas simples. Es preferible simular *n* variables aleatorias secuencialmente resultantes de las *n* univariantes condicionales en vez de generar un vector n-dimensional en un único paso usando la distribución total conjunta, que en general es muy compleja (Walsh 2002).

Para presentar el mecanismo del muestreo Gibbs, se considera una variable aleatoria de 2 componentes $(x,y)$, y se desea calcular $p(x)$ y/o $p(y)$. La forma tradicional es integrar la densidad conjunta $p(x) = \int p(x,y)dy$, que, como se vio anteriormente, suele ser costosa computacionalmente. Para resolver este problema se hace uso de la técnica de muestreo. La idea del muestreo es que es más fácil considerar una secuencia de distribuciones condicionales $p(x|y)$ y $p(y|x)$ que obtener la distribución conjunta $p(x,y)$. De esta forma, se puede pensar en un proceso markoviano donde la

distribución condicional $p(x|y)$ juegue el rol de probabilidad de transición del sistema, y bajo ciertas condiciones de ergodicidad, que en general se satisfacen, $p(x|y)$ converge a $p(x)$. El problema consiste en generar el proceso de Markov cuya probabilidad estacionaria sea $p(x)$. A continuación se enunciará como se puede lograr.

La idea del muestreo vista en MCMC, en la cual se muestrea sobre una variable y luego sobre la otra es aplicada en este caso. De este modo, el muestreo comienza asignando un valor inicial $y_0$ a $y$ y obteniendo un $x_0$ mediante la generación de una variable aleatoria desde la distribución condicional $p(x|y=y_0)$. Luego usa ese valor $x_0$ para generar un nuevo valor de $y$, el $y_1$, que sigue la distribución condicional $p(y|x=x_0)$. El muestreo procede de la siguiente forma:

$$x_i \sim p(x|y=y_{i-1})$$

$$y_i \sim p(y|x=x_i)$$

Se repite este proceso *k* veces, y se genera una secuencia Gibbs de longitud $k$, donde los $(x_i, y_i)$ con $i=0,1,...k$ son tomados como muestras que siguen una distribución total conjunta $p(x,y)$. Aplicando Monte Carlo se obtiene $p(x)=\int p(x,y)dy \simeq \frac{1}{n}\sum_{i=1}^{n} p(x_i, y_i)$

El muestreo es extendido de la misma forma cuando están involucradas más de dos variables. Sea *D* los datos del sistema y $A=\{a_1,....,a_n\}$ un conjunto de $n$ variables aleatorias independientes cuya distribución de probabilidad conjunta $P(a_1,...,a_n, D)$ es desconocida. En este caso, cada nuevo valor de $A$ se obtiene a través de un proceso iterativo que sólo requiere generar muestras de distribuciones cuya dimensión es menor que *n* y que en la mayoría de los casos tienen una forma más sencilla que la de distribución condicional completa $P(A|D)$. La idea del muestreo Gibbs es obtener muestras de cada una de las componentes de $A$ dejando en cada tiempo todas las restantes componentes fijas. De esta manera el sistema se transforma en univariable, y por consiguiente muy simple de resolver. En particular, este último caso de multivariables es el que interesa en esta tesis, debido a que las posiciones de los motivos en las distintas secuencias estarían representado por el vector de incógnitas $A$. En la sección siguiente se detalla el algoritmo del muestreo Gibbs para la resolución del caso biológico.

### 2.6.2 Algoritmo

Sean $D$ los datos observables del sistema y $A=\{a_1,....,a_n\}$ las variables ocultas. Por consiguiente $P(a_1,...,a_n, D)$ es la distribución buscada. El algoritmo, que se muestra en la fig.2.1, comienza suponiendo una muestra inicial $A^{(0)}=(a_1^{(0)},...,a_n^{(0)})$ .

**Algoritmo de muestreo Gibbs**

**Datos:** $A$ variables no observadas, $D$ datos observados, $n$ cantidad de iteraciones

**Resultado:** La distribución condicional $P(A|D)$

**Algoritmo**

Inicializar las variables no observadas $a_i^{(0)}$ , $i=1,2,...,n$

Repetir para $t=1,...,n$

Repetir para cada una de las variables no observadas $k=1,2,...,n$ hacer

Dejar fijas las variables $1,...,k-1,k+1,...,n$

Calcular la distribución condicional conjunta $P(a_k |a_1^{(t)},...,a_{k-1}^{(t)},a_{k+1}^{(t-1)},...,a_n^{(t-1)},D)$

Generar un muestreo $a_k^{(t)}$ sobre la distribución conjunta.

fin repetir

Calcular la distribución condicional $P(A^{(t)}|D)$ dados los estimadores $(a_1^{(t)},...,a_n^{(t)})$

fin repetir

Retornar la distribución condicional de los estimadores.

Fig.2.1. Algoritmo de muestreo Gibbs

En la *t-ésima* iteración el valor de la *k-ésima* variable sigue una distribución condicional completa $P(a_k \mid A^{[-k]}, D)$ siendo $A^{[-k]} = \{a_j,\ j \neq k\}$ el vector de todas las variables salvo la *k-ésima*, y *D* el conjunto de datos observados, el muestreo resulta:

$$a_k^{(t)} \sim \mathrm{P}(a_k \mid a_1^{(t)},...,a_{k-1}^{(t)}, a_{k+1}^{(t-1)},...,a_n^{(t-1)}, D) \text{ con } k = 1,2,...,n$$

Esta distribución condicional completa debe ser sencilla de muestrear.

A continuación se ejemplifica para *n* = 3.

Sea $A = (a_1, a_2, a_3)$ con condiciones iniciales $A^{(0)} = (a_1^{(0)}, a_2^{(0)}, a_3^{(0)})$.
En la *t-ésima* iteración el muestreo Gibbs genera para cada componente de $A$ un valor obtenido de la siguiente manera:

a) $a_1^{(t)} \sim \mathrm{P}(a_1 \mid a_2^{(t-1)}, a_3^{(t-1)}, D)$

b) $a_2^{(t)} \sim \mathrm{P}(a_2 \mid a_1^{(t)}, a_3^{(t-1)}, D)$

c) $a_3^{(t)} \sim \mathrm{P}(a_3 \mid a_1^{(t)}, a_2^{(t)}, D)$

Se observa que en cada paso la variable a muestrear no es considerada en el condicionamiento, y la nueva observación es usada en el próximo paso como una "mejor" estimación del valor anterior. Después de estos pasos se obtiene una actualización $A^{(t)} = (a_1^{(t)}, a_2^{(t)}, a_3^{(t)})$ a partir del anterior $A^{(t-1)} = (a_1^{(t-1)}, a_2^{(t-1)}, a_3^{(t-1)})$. Una cuestión en este punto es preguntar si realmente se está en presencia de una cadena de Markov. Para ello se debe demostrar que los $A^{(t)} = (a_1^{(t)}, a_2^{(t)}, a_3^{(t)})$ en la *t-ésima* iteración sólo depende de los $A^{(t-1)} = (a_1^{(t-1)}, a_2^{(t-1)}, a_3^{(t-1)})$ de la iteración precedente y no de los anteriores $A^{(j)},\ j = 0,....,(t-2)$. En la fig.2.2 se plantea esta prueba, donde se demuestra que este $A^{(t)}$, generado como se describió, constituye un estado de una cadena de Markov. Con esta comprobación, se puede asegurar que el proceso del muestreo Gibbs genera una cadena de Markov con estados $\{A^{(m)} = (a_1^{(m)},..., a_n^{(m)}), m = 0,1,...,t,...\}$ con probabilidad de transición de estados $P(A^{(t+1)} \mid A^{(t)}, D)$ desde un estado $A^{(t)}$ a otro $A^{(t+1)}$, la cual está definida de la siguiente manera:

$$P(A^{(t+1)} \mid A^{(t)}, D) = \prod_{i=1}^{k} \mathrm{P}(a_i^{(t+1)} \mid a_1^{(t+1)},..., a_{i-1}^{(t+1)}, a_{i+1}^{(t)},..., a_n^{(t)}, D)$$

Después de una cantidad suficiente de iteraciones se llega a la distribución estacionaria. Bajo ciertas condiciones de ergodicidad, donde se asegura la representatividad de la muestra, la cadena $(A^{(0)}, A^{(1)},..., A^{(m)})$ converge a una distribución de equilibrio que no depende de los valores iniciales. Por construcción los valores $A = (a_1,..., a_n)$ de cada componente de la cadena es un muestreo del estimador de Monte Carlo. La distribución de estos estimadores es la distribución de probabilidad buscada $P(A \mid D)$. Ciertamente la distribución conjunta $P(a_1^{(t)},..., a_n^{(t)}, D)$ converge a $P(a_1,..., a_n \mid D) = P(A \mid D)$ cuando $t \to \infty$.

La técnica del muestreo Gibbs está bien preparada para sistemas que requieren un bajo consumo de memoria (Ide *et al*. 2001 [27]) debido a la simplicidad del método, y para la inferencia de procesos que deben producir resultados en un tiempo acotado (Ramos *et al.* 2002 [52]).

Como se verá en la sección siguiente la formulación del Muestreo Gibbs es muy similar al algoritmo Expectation-Maximization. Ello permitirá descubrir las ventajas del Muestreo Gibbs frente a este método.

**Demostración:** $A^{(t)}$ ***es un estado de una cadena de Markov***

Para demostrar que un $A^{(t)}$ obtenido por el muestreo Gibbs es un estado de una cadena de Markov, se debe demostrar que el estado t-ésimo de $A^{(t)}$ sólo depende del estado anterior $A^{(t-1)}$.

Si se está en la segunda iteración, se debe demostrar que $P(A^2 \mid A^1, A^0) = P(A^2 \mid A^1)$.

Para simplificar la demostración se considera que cada parámetro tiene 2 componentes $A = (a_1, a_2)$

Sea $$P(A^2 \mid A^1, A^0) = P(a_1^2, a_2^2 \mid a_1^1, a_2^1, a_1^0, a_2^0)$$

por propiedad de la probabilidad condicional

$$P(A^2 \mid A^1, A^0) = \frac{P(a_1^2, a_2^2, a_1^1, a_2^1, a_1^0, a_2^0)}{P(a_1^1, a_2^1, a_1^0, a_2^0)}$$

introduciendo una simplificación de la notación: $A^0 = a_1^0, a_2^0$ y $A^1 = a_1^1, a_2^1$, resulta

$$P(A^2 \mid A^1, A^0) = \frac{P(a_1^2, a_2^2, A^1, A^0)}{P(A^1, A^0)} = \frac{P(a_2^2 \mid a_1^2, A^1, A^0) P(a_1^2, A^1, A^0)}{P(A^1, A^0)}$$

por construcción del muestreo Gibbs $P(a_2^2 \mid a_1^2, A^1, A^0) = P(a_2^2 \mid a_1^2, A^1)$

$$P(A^2 \mid A^1, A^0) = \frac{P(a_2^2 \mid a_1^2, A^1) P(a_1^2, A^1, A^0)}{P(A^1, A^0)}$$

reemplazando $P(a_2^2, A^1, A^0) = P(a_2^2 \mid A^1, A^0).P(A^1, A^0)$

$$P(A^2 \mid A^1, A^0) = \frac{P(a_2^2 \mid a_1^2, A^1) P(a_1^2 \mid A^1, A^0).P(A^1, A^0)}{P(A^1, A^0)} = P(a_2^2 \mid a_1^2, A^1) P(a_1^2 \mid A^1, A^0)$$

por construcción del muestreo Gibbs $P(a_1^2 \mid A^1, A^0) = P(a_1^2 \mid A^1)$

$$P(A^2 \mid A^1, A^0) = P(a_2^2 \mid a_1^2, A^1) P(a_1^2 \mid A^1)$$

por propiedad de la probabilidad condicional

$$P(A^2 \mid A^1, A^0) = \frac{P(a_2^2, a_1^2, A^1) P(a_1^2 \mid A^1)}{P(a_1^2, A^1)} = \frac{P(a_2^2, a_1^2, A^1) P(a_1^2, A^1)}{P(a_1^2, A^1) P(A^1)}$$

simplificando $$P(A^2 \mid A^1, A^0) = \frac{P(a_2^2, a_1^2, A^1)}{P(A^1)}$$

donde el segundo término es $P(a_1^2, a_2^2 \mid A^1) = P(A^2 \mid A^1)$

con lo cual se demuestra que $P(A^2 \mid A^1, A^0) = P(A^2 \mid A^1)$

Fig.2.2. Demostración $A^{(t)}$ es un estado de una cadena de Markov

## 2.7 Expectation Maximization (EM)

El muestreo Gibbs se puede considerar como un proceso estocástico equivalente al *Expectation-Maximization* (EM, Demspter *et al*. 1977) en el sentido que tiene una primer fase de actualización y una segunda de maximización. A pesar que en esta tesis no se utilizará el algoritmo EM tal cual es, una explicación de su funcionamiento es interesante y permite sentar las bases para entender el Muestreo Gibbs.

El EM es un algoritmo iterativo de carácter general para la estimación MLE de parámetros, aplicable a casos en que algunos datos del sistema en análisis no sean observados. Dado un modelo estadístico determinado por parámetros $q$, las secuencias observadas $X_{obs}$ y los datos no observables $X_{mis}$, el objetivo es encontrar el mejor modelo bajo el criterio MLE, según (2.5).

El cálculo del likelihood de los datos con parámetros $A$ se obtiene a partir de las probabilidades condicionales de los datos no observables $X_{mis}$, con el objetivo de encontrar el modelo que maximice

$$P(X_{obs} \mid q) = \prod_{X_{mis}} P(X_{obs}, X_{mis} \mid q)$$

Para obtener la solución a este problema se debe realizar un proceso iterativo, donde en cada repetición se obtenga una estimación de los parámetros. La sucesión de estimadores obtenidos con este método deberá generar valores de likelihood crecientes. Esta medida crece en cada iteración, y se repite hasta que el crecimiento sea despreciable.

El proceso EM se explica a continuación.

Si se dispone en la iteración *t-ésima* un modelo válido de parámetros $q^{(t)}$, se desea estimar un nuevo y mejor modelo con parámetros $q^{(t+1)}$.

Sea $P(X_{obs}, X_{mis} \mid q) = P(X_{mis} \mid X_{obs}, q) P(X_{obs} \mid q)$ la probabilidad de los datos dado el modelo. Entonces el log-likelihood resulta:

$$\log P(X_{obs} \mid q) = \log P(X_{obs}, X_{mis} \mid q) - \log P(X_{mis} \mid X_{obs}, q)$$

Multiplicando por $P(X_{mis} \mid X_{obs}, q^{(t)})$ y sumando sobre todos los $X_{mis}$ resulta

$$\log P(X_{obs}\,|q) = \sum_{X_{mis}} P(X_{mis}\,|\,X_{obs},q^{(t)})\log P(X_{obs},X_{mis}\,|q) - \sum_{X_{mis}} P(X_{mis}\,|\,X_{obs},q^{(t)})\log P(X_{mis}\,|\,X_{obs},q)$$

Si se designa $Q(q\,|\,q^{(t)}) = \sum_{X_{mis}} P(X_{mis}\,|\,X_{obs},q^{(t)})\log P(X_{obs},X_{mis}\,|q)$ (2.13)

resulta

$$\log P(X_{obs}\,|\,A) = Q(q\,|\,q^{(t)}) - \sum_{X_{mis}} P(X_{mis}\,|\,X_{obs},q^{(t)})\log P(X_{mis}\,|\,X_{obs},q) \qquad (2.14)$$

Como el objetivo es iterar para maximizar el $\log P(X_{obs}\,|q)$, se desea que la diferencia entre éste y el obtenido en el paso *t-ésimo* sea positiva.

$$\log P(X_{obs}\,|q) - \log P(X_{obs}\,|\,q^{(t)}) > 0$$

Usando la ecuación (2.14) para calcular la diferencia resulta:

$$\log P(X_{obs}\,|q) - \log P(X_{obs}\,|\,q^{(t)}) =$$

$$= Q(q\,|\,q^{(t)}) - Q(q^{(t)}\,|\,q^{(t)}) + \sum_{X_{mis}} P(X_{mis}\,|\,X_{obs},q^{(t)})\log \frac{P(X_{mis}\,|\,X_{obs},q^{(t)})}{P(X_{mis}\,|\,X_{obs},q)}$$

El último término es la entropía relativa de $P(X_{mis}\,|\,X_{obs},q^{(t)})$ para $P(X_{mis}\,|\,X_{obs},q\,)$, que es siempre un valor no negativo, por lo que resulta

$$\log P(X_{obs}\,|q) - \log P(X_{obs}\,|\,q^{(t)}) \geq Q(q\,|\,q^{(t)}) - Q(q^{(t)}\,|\,q^{(t)})$$

La igualdad se cumple si $q = q^{(t)}$ o si $P(X_{mis}\,|\,X_{obs},q^{(t)}) = P(X_{mis}\,|\,X_{obs},q)$ para otro $q \neq q^{(t)}$. En especial si elegimos un $q$ tal que $q^{(t+1)} = \arg\max_{q} Q(q\,|\,q^{(t)})$ la diferencia será ciertamente positiva.

La función $Q$ en (2.13) es un promedio del logaritmo de $P(X_{obs},X_{mis}\,|q)$ sobre la distribución de probabilidades de $X_{mis}$ obtenida con los parámetros $q^{(t)}$. Además, $Q$ representa una estimación del MLE buscado.

En este método de ajuste y maximización no es necesario maximizar $Q$ en un sentido estricto, simplemente se va encontrando valores de $Q(q^{(t+1)}|q^{(t)})$ que son mayores a su anterior $Q(q^{(t)}|q^{(t)})$. Cuando la diferencia entre dos $Q$ consecutivos sea despreciable se habrá llegado a un máximo local sobre $q$.

Aunque EM garantiza convergencia, esta puede ser a un máximo local, por lo que se recomienda repetir el proceso varias veces, Durbin *et al*. 2002.

Como conclusión se puede decir que el método EM y el muestreo Gibbs se construyen sobre la misma base estadística. Sin embargo, en el EM se calcula el likelihood de los datos dado el modelo mientras que en el muestreo Gibbs se pretende calcular la probabilidad posterior del alineamiento de los motivos dados los datos. Además, en el EM al considerar todas las posibilidades con todas las secuencias, su complejidad crece exponencialmente con el agregado de nuevas secuencias. Por el contrario, en el muestreo Gibbs al considerar una única secuencia por vez se obtiene una complejidad lineal con respecto a la cantidad de secuencias. Los métodos EM son determinísticos y pueden quedar atrapados en máximos locales, los cuales pueden ser evitados por el muestreo estocástico del Gibbs. A partir de la naturaleza estocástica y la idea de considerar una variable por vez con el resto de las variables fijas, el muestreo Gibbs permite construir modelos más realistas y sofisticados, como el hecho de localizar en forma simultánea patrones múltiples y patrones repetitivos.

En la sección siguiente se presenta una aplicación del método EM a la búsqueda de motivos en secuencias biológicas.

Con relación al problema central de esta tesis, el algoritmo EM se utiliza en el core del paquete MEME [MEME] . Las características del MEME se enuncian en la sección 3.6; MEME es un programa de acceso público que permite localizar motivos conservados en secuencias protéicas o de ADN, y se utiliza en esta tesis para comparar el rendimiento del programa GibbsSM.

## 2.8 Modelado estadístico de secuencias biológicas

### 2.8.1 Motivación

Tal como se expresó en el Capítulo anterior las moléculas que comparten un antepasado común heredan similitudes de la estructura primaria del antepasado. Esto es conocido como conservación de la estructura primaria en una familia. De esta manera, en la práctica, las secuencias biológicas se reúnen por similaridades en familias funcionales. En cada caso las secuencias involucradas normalmente mantienen la misma función o una función relacionada. Muchos de los métodos de análisis de secuencias ya presentados se basan en identificar la relación de una secuencia individual con una familia de secuencias. Ciertamente, identificar que una secuencia pertenece a una familia permite realizar inferencias acerca de su función. Si ya se dispone de un conjunto de secuencias pertenecientes a una familia, se puede hacer una búsqueda en una base de datos para hallar más miembros haciendo una alineación local con cada uno de las secuencias conocidas de la familia. Como resultado de este proceso es posible identificar zonas preservadas entre las secuencias que evidencian una relación entre ellas, los dominios. Sin embargo, la búsqueda de a pares con cualquiera de los miembros, además de ser un método engorroso, puede no encontrar secuencias relacionadas distantemente. Un método alternativo de búsqueda es usar las características estadísticas del conjunto completo de secuencias. Esta idea lleva a realizar el análisis de las alineaciones de las secuencias descubriendo sus propiedades estadísticas o determinísticas para proponer modelos que se ajusten a las mismas. Utilizar un modelo estadístico que identifica a la familia de secuencias hace posible descubrir hipótesis sobre las relaciones entre ellas e identificar nuevas secuencias vinculadas. De esta forma se genera información que podrá ser utilizada para el diseño de experiencias en el laboratorio.

Por la propia forma en que se codifica la información biológica, muchos de los problemas relacionados con su análisis requieren algún tipo de modelo adecuado para las secuencias biológicas. Los modelos más utilizados son los frecuentistas, las cadenas de Markov y los Modelos Ocultos de cadenas de Markov (Hidden Markov Model - HMM). En esta sección se presenta como modelar probabilísticamente las secuencias biológicas y el modelado de familias de secuencias biológicas a través de perfiles. Dentro de este último tipo, se hará hincapié en el HMM por ser uno de los modelos más importante utilizados en Bioinformática.

### 2.8.2 Modelo probabilístico

Frecuentemente en Biología molecular se asume que en cada secuencia biológica un residuo *a* ocurre al azar con una probabilidad $q_a$ en forma independiente de todo otro residuo e idénticamente distribuido (i.i.d.). Dicha probabilidad $q_a$ puede ser estimada a partir de un conjunto considerable de secuencias denominado de *entrenamiento*. De este conjunto se conocen sus propiedades, y por lo tanto se podrá ajustar un modelo. Cuanto más secuencias se dispongan mejor va a ser la aproximación de los parámetros del modelo. Una vez ajustado el modelo se podrá determinar si una secuencia $x_{1,}\ x_{2,\ldots}\ x_n$ pertenece o no a dicha familia calculando la probabilidad conjunta como el producto de los valores individuales:

$$q_{x_1} * q_{x_2} * \ldots * q_{x_n} = \prod_{i=1}^{i=n} q_{x_i}$$

A continuación se ejemplifica sobre un conjunto de secuencias de ADN. Se desea encontrar el modelo probabilístico $M$ que las identifica. Para las consideraciones probabilísticas es lo mismo considerar varias secuencias o una única compuesta por la encadenación de todas, una larga secuencia de residuos $D = \{x_1, x_2, \ldots.. x_N\}$ de longitud *N*, con $x_i \in Alfa$ , $Alfa = \{A, C, G, T\}$. Por lo cual el modelo $M$ resulta tener 4 parámetros $p_A, p_C, p_G, p_T$ cuyos valores se desean estimar. Lo único que se sabe es que $p_A + p_C + p_G + p_T = 1$. Además, bajo el modelo de independencia, la probabilidad de observar la secuencia *D* resulta ser:

$$P(D \mid p_A, p_C, p_G, p_T) = \prod_{x \in Alfa} p_x^{n_x} = p_A^{n_A} p_C^{n_C} p_G^{n_G} p_T^{n_T} \qquad (2.15)$$

donde $n_x$ es la cantidad de residuos *x* que aparece en la secuencia $D$. Para hallar los valores de los parámetros se puede aplicar el teorema de Bayes, ecuación $(2.1)$, y estimar por MLE. Sea $p_A, p_C, p_G, p_T$ los parámetros del modelo *M*, la probabilidad a posteriori del modelo se calcula de la siguiente manera:

$$P(M \mid D) = \frac{\prod_{x \in Alfa} p_x^{n_x} P(M)}{P(D)} \qquad (2.16)$$

aplicando el logaritmo negativo se obtiene:

$$-\log P(M \mid D) = -\sum_{x \in A} n_x \log p_x - \log P(M) + \log P(D) \qquad (2.17)$$

Si se supone una distribución a priori uniforme sobre los parámetros, la estimación del MLE resulta:

$$p_{MLE} \equiv \max - \sum_{x \in Alfa} n_X \log p_X \qquad (2.18)$$

resolviendo por optimización del lagrangiano

$$L = -\sum_{x \in Alfa} n_x \log p_x - l\,(1 - \sum_{x \in A} p_x)$$

resulta

$$p^*_x = \frac{n_x}{N} \text{ para todo } x \in Alfa \qquad (2.19)$$

Esta estimación frecuentista es natural cuando $N$ es grande. Pero, ¿qué sucede si $N$ es pequeño? Puede ser que algunos residuos no estén presentes y por lo tanto su $p_x$ sea *0* y, en consecuencia, la probabilidad a priori no resulta uniforme, contradiciendo la hipótesis enunciada. En orden de incluir una información a priori que refleje estos casos se propone una distribución *Dirichlet* sobre los parámetros [58]. El uso de la distribución Dirichlet en Biología molecular está muy difundido. La distribución Dirichlet es una distribución discreta que posee una gran cantidad de parámetros y por lo tanto es muy versátil para modelar una gran cantidad de escenarios de conocimiento a priori respecto a los parámetros del modelo bajo análisis. En este sentido se asemeja con la distribución *gama* la cual también es muy usada en Biología molecular para el modelado de conocimiento a priori de parámetros continuos.

La función de densidad de una variable aleatoria Dirichlet tiene la forma $f_0(p) \propto \prod_{j=1}^{d} p_j^{a_j - 1}$

Reemplazando en (2.18) $P(M)$ por esta función de densidad, la probabilidad a posteriori resulta:

$$-\log P(M \mid D) = -\sum_{x \in Alfa} (n_x + a_X - 1) \log p_x + \log P(D)$$

El problema de optimización MAP es muy similar al resuelto recién excepto que los $n_X$ son reemplazados por $(n_X + a_x - 1)$, por lo tanto los parámetros del modelo *M* resultan ser:

$$p^*_X = \frac{(n_X + a_X - 1)}{N + a - |Alfa|} \qquad (2.14)$$

Se concluye que el efecto de la distribución de *Dirichlet* tomada como distribución a priori es equivalente a agregar *pseudocounts* para las cantidades observadas de residuos cuando no se dispone de un número suficientemente grande de ellos. Se verá en el capítulo siguiente que este razonamiento se aplica al contemplar las distribuciones a priori de los residuos al aplicar el muestreo Gibbs para la detección de motivos en secuencias biológicas.

### 2.8.3 Modelado de familias de secuencias: perfiles y HMM

Un *perfil estadístico* es un modelo simplificado que se utiliza para caracterizar una familia de secuencias. El método más usado para determinar perfiles es el que enunció Gribskov *et al*. en 1990 [21]. Este método requiere una alineación múltiple a partir del cual se obtienen los perfiles calculando la distribución de frecuencias de cada residuo en cada posición de la alineación. Estos valores se usan para obtener puntuaciones del perfil con otras secuencias. Cuando el puntaje está por encima de cierto umbral se considera que dicha secuencia puede pertenecer a la familia.

Para ejemplificar se presenta la siguiente alineación de 5 secuencias de proteínas.

```
Posición    1 2 3 4 5
            C C G T L
            C G H S V
            G C G S L
            C G G T L
            C C G S S
```

El perfil estadístico de esta alineación utilizando el método de Gribskov resulta ser:

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Prob(C) | 0.8 | 0.6 | - | - | - |
| Prob(G) | 0.2 | 0.4 | 0.8 | - | - |
| Prob(H) | - | - | 0.2 | - | - |
| Prob(S) | - | - | - | 0.6 | 0.2 |
| Prob(T) | - | - | - | 0.4 | - |
| Prob(L) | - | - | - | - | 0.6 |
| Prob(V) | - | - | - | - | 0.2 |

Según este perfil, la probabilidad que el residuo C aparezca en la primer posición de la alineación es 0.8, y la de G en la segunda posición es 0.4.

Dada una familia de secuencias modelada a través de un perfil estadístico, la probabilidad que otra secuencia pertenezca a la familia es el producto de las probabilidades del residuo dado por el perfil. Por ejemplo, la probabilidad de la secuencia CGGSV es $0.8 * 0.4 * 0.8 * 0.6 * 0.2 = 0.031$ .

Ya que la multiplicación es computacionalmente cara y propensa a errores de punto flotante, se realiza una transformación logarítmica. De esta manera el puntaje de una secuencia se calcula tomando los logaritmos de las probabilidades de cada residuo y luego se los suma. Resulta que el puntaje de CGGSV es $\log(0.8)+\log(0.4)+\log(0.8)+\log(0.6)+\log(0.2) = -3.48$

En la práctica, los perfiles tienen en cuenta otros factores tales como establecer penalizaciones por inserciones o borrados; dar más peso a un residuo probable de aparecer en una posición estructuralmente importante que a uno que aparece en una posición estructuralmente insignificante, entre otros. Estos refinamientos son necesarios tenerlos en cuenta a la hora de crear buenos modelos. Pero al considerar estas restricciones se introducen más parámetros que deben calcularse al construir el perfil. Desgraciadamente en estos casos, mediante esta metodología, los cálculos deben ser hechos por prueba y error. Estas limitaciones pusieron en escena un nuevo tipo de perfiles basado en un modelado estadístico, el Modelo Oculto de cadenas de Markov (HMM). A continuación se analizan estos modelos.

## HMM - Hidden Markov Models

El modelado de familias de secuencias por HMM fue introducido en Bioinformática en 1994 por Krogh *et al*. [31] realizando una extensión de las técnicas utilizadas clásicamente en reconocimiento de patrones de voz. Desde entonces este método se ha convertido en uno de los más populares. Los HMM ofrecen un método estadístico sistemático para estimar los parámetros del modelo. Es una especie de perfil estadístico, y como tal es construido analizando la distribución de los residuos en las secuencias de entrenamiento. A diferencia del perfil de Gribskov, un HMM tiene una topología más compleja y permite el modelado de sistemas con información oculta. En el ámbito de la Bioinformática se lo ha implementado en varios programas computacionales, tales como el HMMER, utilizado en la construcción de bases de datos de perfiles como la Pfam [PFAM].

La metodología de HMM permite el modelado de perfiles complejos, y puede ser pensado como una máquina de estados finitos. Las secuencias biológicas son los datos observables y los estados de la máquina de estados finitos es la información no visible del sistema. Un HMM está definido por un conjunto finito de estados, un alfabeto de símbolos, una matriz de probabilidades de transición de un estado a otro, y una matriz de probabilidades de emitir un símbolo estando en un estado.

Una vez construido, un HMM genera una secuencia biológica emitiendo residuos cuando va pasando a través de los estados. Cada estado tiene una tabla de probabilidades de emisión de residuos similar a la descripta para un perfil, y tablas de probabilidades de transiciones de un estado a otro.

En la figura 2.3, extraída de [SHAMIR], se muestra una topología muy popular de un modelo

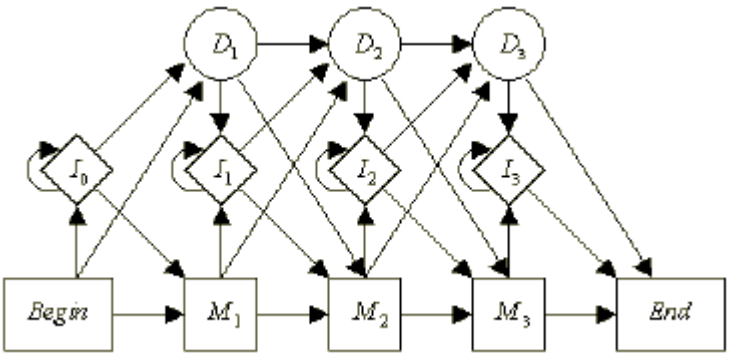


Fig.2.3 Topología típica de un modelo HMM

HMM. En la misma hay 3 figuras diferentes: círculos, cuadrados y rombos que representan estados. Los *cuadrados* son los estados principales, denominados estados de *match* (coincidencias), y modelan la distribución de probabilidad que surge del alineamiento múltiple; los *rombos* son estados de *inserciones* y emiten un residuo como resultado de una inserción; y los *círculos* son estados de *remociones* de residuos. Las transiciones de un estado a otro progresan de izquierda a derecha a través del modelo, con excepciones que puedan tener lazos sobre los estados de inserción (rombos).

Un camino (*path*) $p = p_1, p_2, ..., p_L$ en el modelo $M$ es una sucesión de estados. Dada una secuencia $S = s_1, ...., s_n$ se pueden definir las probabilidades de transición y emisión de la siguiente manera:

$a_{kl} = P(p_i = l \mid p_{i-1} = k)$ probabilidad de transición del estado $k$ al $l$

$e_k(b) = P(s_i = b \mid p_i = k)$ probabilidad de emisión del residuo $b$ en le estado $k$

A partir de estos valores se puede calcular la probabilidad de que una secuencia $S$ sea generada por el modelo $M$ dado el camino $p$ , y está dada por:

$$P(S \mid p) = (a_{p_0, p_1}) \prod_{i=1}^{L} (e_{p_i}(s_i) a_{p_i, p_{i+1}})$$

La figura 2.4, extraída de Rachel Karchin (1999) [28], muestra un posible modelo de HMM para un conjunto de secuencias. El mismo está representado por un modelo de probabilidades: la

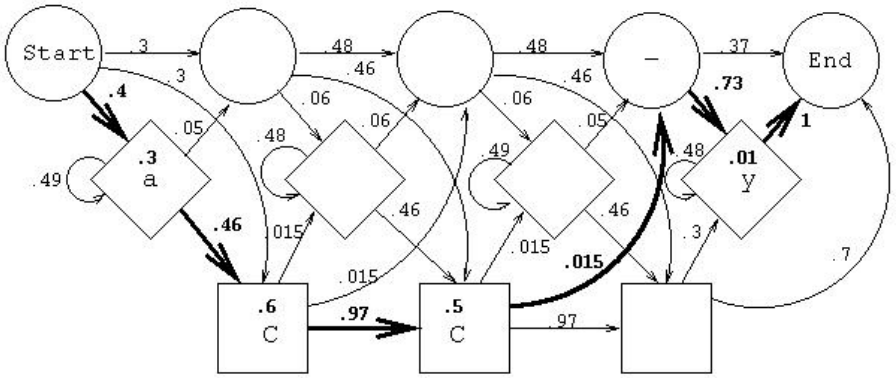


Fig.2.4 Un HMM derivado de un conjunto de secuencias.

probabilidad que un residuo ocurra en un estado particular y las probabilidades de transición de un estado a otro. La probabilidad que el modelo genere la secuencia ACCY, según la fórmula anterior, está dada por:

$$0.4 * 0.3 * 0.46 * 0.6 * 0.97 * 0.5 * 0.015 * 0.73 * 0.01 * 1 = 1.76 \times 10^{-6}.$$

Este cálculo es fácil si los estados del camino a recorrer son conocidos, como en el ejemplo anterior, pero en modelos reales existen muchos caminos para generar la misma secuencia. De este modo, la probabilidad de una secuencia de residuos resulta ser la suma de las probabilidades de todos los caminos posibles. Desafortunadamente este cálculo de fuerza bruta es computacionalmente impracticable, excepto en el caso de secuencias muy cortas.

En este contexto se plantean tres cuestiones que resumen el funcionamiento de un HMM:

- ✓ ¿Cuál es el camino de estados más probable que se recorre al representar una secuencia con el modelo? Este problema de Decodificación es denominado *Decoding* por [SHAMIR](2002) y *Alignment problem* según Rabiner (1989) [51]
- ✓ ¿Cuál es la probabilidad de que una secuencia de residuos *S* sea generada por el modelo? Este problema de Evaluación es denominado *Scoring* según Rabiner (1989) [51]
- ✓ ¿Cuáles son los parámetros que mejor ajustan el modelo al conjunto de datos? Este problema de Estimación es denominado *Training* según Rabiner (1989)[51]

*Decoding*

Un problema para el primer planteo puede ser identificar una región codificante en una secuencia de ADN. Si el camino más probable pasa por una subsecuencia de estados perteneciente a dicho tipo, se puede considerar que la región codificante se encuentra en la subsecuencia. Para resolver el problema de la *decodificación* se recurre a un algoritmo que tenga como entradas el modelo HMM y la secuencia dada, y permita hallar "la serie de estados ocultos más probables que se transitaron cuando dicha secuencia fue obtenida". Estos estados son aquellos que maximicen la probabilidad de observar la secuencia en dicho modelo. El algoritmo de *Viterbi* es el más utilizado para solucionar este problema. Es un algoritmo de programación dinámica donde se contempla cada estado del modelo, en un tiempo dado, con sus probabilidades de emisión y las cantidades calculadas en un tiempo anterior.

El modelo se recorre desde el comienzo hasta el final y en cada unidad de tiempo se almacena el estado cuyo valor fue máximo, siendo éste el estado más probable.

*Scoring*

En este punto, el problema que se plantea es querer clasificar una nueva proteína. Para ello se puede construir un HMM para cada familia de proteínas conocidas y utilizarlo para calcular la probabilidad de pertenencia de la nueva proteína. Asignando la proteína a la familia en cuyo modelo obtenga mayor probabilidad. Para encarar el problema de *evaluación* habría que recorrer todos los posibles caminos que pudieran haber generado la secuencia dada y calcular cada una de sus probabilidades. Se debe tener en cuenta que el número de caminos crece exponencialmente con la longitud de la secuencia por lo cual este cálculo es muy costoso. Sin embargo, se puede hacer uso del algoritmo recursivo *forward* que permite realizar el cálculo de forma rápida y eficaz.

La complejidad de estos algoritmos, Viterbi y forward, es $O(M*T)$ donde $M$ es la cantidad de transiciones con probabilidad distinta de cero y $T$ es la longitud de la secuencia de entrada. $M$ puede ser a lo sumo $S*S$ donde $S$ es la cantidad de estados del modelo, pero usualmente es bastante menor, puesto que en general la matriz de probabilidades de transición tiene pocos elementos distintos de cero.

*Training*

Por último, tanto para resolver el primer problema como el segundo se debe configurar el modelo, y por lo tanto se debe definir la arquitectura del HMM y estimar sus parámetros. Para poder estimar los parámetros de un HMM se necesita: la topología del HMM (lista de todos los nodos y transiciones posibles) y, los datos (cuantos más, mejor).

El algoritmo de *estimación* de parámetros más popular se conoce con el nombre de *Baum-Welch.* Es un algoritmo de tipo EM, que permite calcular las probabilidades de emisión y transición en el caso de no conocerse los estados. También suelen utilizarse aproximaciones bayesianas del tipo MAP interesantes en tanto que permiten describir las suposiciones iniciales sobre el modelo con una distribución a priori adecuada.

Para determinar el modelo HMM se utilizan secuencias de entrenamiento, de las cuales se conocen sus propiedades y por lo tanto se puede determinar un perfil ajustado al grupo. Cuando los caminos de todas las secuencias de entrenamiento son desentrañados, las probabilidades de emisión y

transición pueden ser calculadas. Sin embargo, los caminos entre estados siguen siendo desconocidos. Encontrar el mejor modelo ajustado al conjunto de entrenamiento es un problema de optimización que debe ser resuelto por métodos iterativos. La idea es encontrar los parámetros del modelo que maximicen la probabilidad de todas las secuencias del conjunto de entrenamiento. Los parámetros son re-estimados después de varias iteraciones para calcular un puntaje para cada secuencia de entrenamiento de tal manera de ir logrando los puntajes más altos posibles, [3][15][31][51] .

*Ventajas de los HMM*

Una de las ventajas de los modelos HMM es la transparencia. De hecho si la arquitectura del modelo está bien diseñada, los usuarios pueden utilizarlo en forma inteligible. El modelo mismo puede contribuir a una mejor comprensión de los procesos.

Otra ventaja es que admiten la incorporación de conocimiento previo a los efectos de diseñar la estructura. Puede inicializarse el modelo cerca de los valores que se consideran adecuados. Ciertamente, puede utilizarse información previa para restringir el dominio de búsqueda en el proceso de estimación.

*Desventajas de los HMM*

Una de las desventajas de los HMM es la suposición de independencia de los estados, que no suele ser del todo cierta. Por otra parte, están presentes los problemas típicos del aprendizaje automático, como es el caer en máximos locales que pueden hacer que el modelo no converja.

Otro problema de los HMM es la velocidad de procesamiento. La mayoría de los procesos que se realizan con HMM requieren realizar enumeraciones. Aún si se utilizan métodos eficientes resultan lentos en comparación con otros métodos.

*Aplicaciones de los HMM en Biología*

Una aplicación común de HMM es clasificar secuencias en una base de datos. La idea es construir un modelo HMM con un conjunto de secuencias de entrenamiento bien conocidas, y luego usar ese modelo para evaluar todas las secuencias en una Base de Datos. Los puntajes así obtenidos son rankeados y un valor umbral se selecciona para separar las secuencias que pertenecen a la clase de las de entrenamiento de las otras.

Otro uso importante del HMM es crear una alineación múltiple de un grupo de secuencias no alineadas. Cada estado de coincidencia en el HMM corresponde a una columna en la alineación. Considerando todas las posiciones en el modelo y los residuos, se puede calcular la probabilidad que cada residuo ocurra en cada posición. Esto produce una tabla de probabilidades para todos los posibles pares de residuo-posición en el modelo. Con esta tabla es posible encontrar el camino de probabilidad más alto a través del modelo para cualquier proteína. Un esquema muy común de HMM es el que se muestra en la figura 2.5, extraída de Dopazo y Valencia (2001) [14]. En esta figura se ejemplifica un alineamiento de cinco secuencias con tres posiciones. Las columnas del alineamiento son los tres estados de match (m1, m2, m3) del sistema. Cada uno de estos estados tiene asociado un vector de probabilidades para cada uno de los residuos posibles (barras correspondientes a las frecuencias observadas de los 20 aminoácidos). También se observan cuatro estados de inserción (i0, i1, i2, i3) y tres estados de borrado (d1, d2, d3). Las flechas representan las posibilidades de transición entre estados.

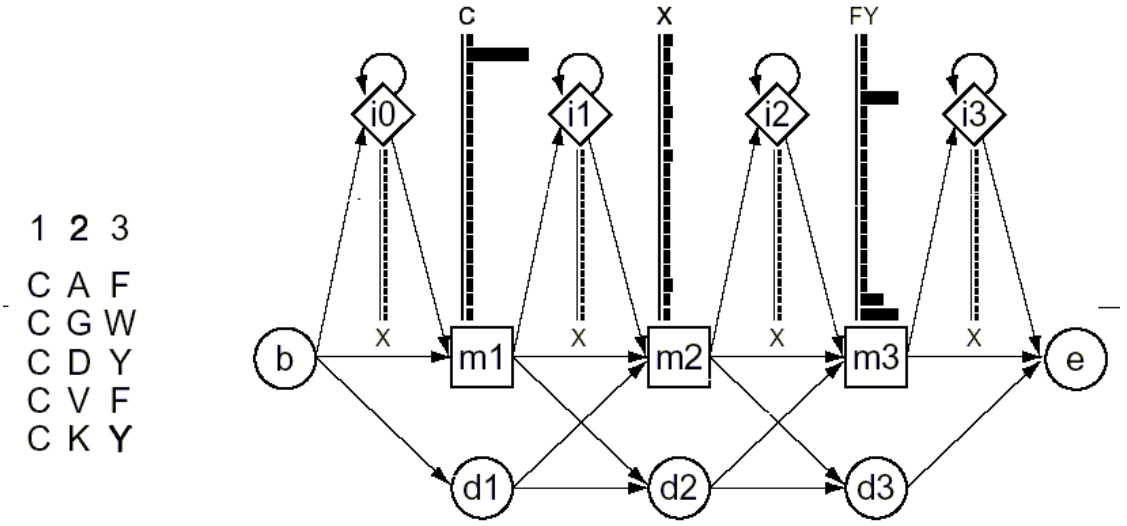


Fig.2.5. Modelo probabilístico HMM de la alineación múltiple adjunta

Modelar familias de secuencias con HMM requieren una gran cantidad de datos, que a veces no se dispone. Computacionalmente un modelo HMM es un generador de secuencias con ciertas propiedades de dependencia que se capturan a partir de datos estimados. Evidentemente, los HMM también son usados para determinar motivos en un conjunto de secuencias biológicas relacionadas, al

igual que el muestreo Gibbs. En particular, el modelado estadístico de secuencias mediante modelos ocultos de Markov (HMM) o muestreo Gibbs permite el diseño de aproximaciones biológicamente significativas sujeto a la disponibilidad de un número adecuado y variado de secuencias. Estas restricciones son especialmente limitantes en el caso de modelos HMM pero salvable en el muestreo Gibbs. A diferencia del modelado HMM, el cual asume una determinada estructura para el proceso de generación de datos, el muestreo Gibbs intenta aproximar la distribución de probabilidad que rige a las secuencias y sus motivos en un proceso iterativo caracterizado por una gran simplicidad algorítmica.

En el Capítulo siguiente se presenta con detalles como resolver el problema de localizar los motivos en un conjunto de secuencias relacionadas mediante la aplicación del muestreo Gibbs, y en el Capitulo 4, Parte experimental, se compara el rendimiento del programa de muestreo Gibbs desarrollado en esta tesis con programas disponibles públicamente. Las secuencias biológicas que se usaron en esta sección fueron extraídas de la base de datos Pfam, la cual utiliza HMM para localiza motivos.

# Capitulo 3: Muestreo Gibbs para motivos

*Una gallina es el medio que usa un huevo para hacer otro huevo*
*Samuel Butler*

## 3.1 Introducción

En el contexto de la Biología computacional, el muestreo Gibbs se aplica a detectar y alinear regiones conservadas o *motivos* en secuencias biológicas [32] [39] [46]. Detectar motivos en un grupo de secuencias de proteínas o ácidos nucleótidos, frecuentemente, sugiere cierta información sobre la estructura, función y evolución molecular de las secuencias analizadas. El muestreo Gibbs permite realizar una alineación local múltiple sin suponer previamente ninguna información sobre los motivos o su ubicación dentro de las secuencias. Por consiguiente, este método determina la ubicación de los motivos sólo a partir de la información intrínseca de las secuencias dadas. La idea subyacente es realizar el análisis de las alineaciones de las secuencias descubriendo sus propiedades estadísticas para proponer modelos que se ajusten a las mismas. En este contexto, los motivos son descriptos por perfiles estadísticos que identifican al grupo de secuencias haciendo posible descubrir hipótesis sobre las relaciones entre ellas.

Una cantidad de algoritmos automatizados para el alineamiento local múltiple han sido desarrollados, y algunos han sido integrados en una variedad de programas de disponibilidad pública. Desafortunadamente, algoritmos rigurosos para encontrar soluciones óptimas han sido computacionalmente caros, y las aproximaciones heurísticas ganan velocidad sacrificando sensibilidad en el caso de motivos altamente variables. La ventaja del método de muestreo Gibbs reside en que es capaz de encontrar una alineación local de $N$ secuencias en un tiempo lineal $N$, permitiendo la detección y optimización simultánea de patrones múltiples y patrones repetitivos. Esta ventaja es lograda por la incorporación de desarrollos en Estadística (vistos en el capítulo anterior), y por el uso de una formulación del problema que modela bien en la línea biológica subyacente. El muestreo Gibbs es reconocido como un método estocástico análogo al EM, ya que tiene dos fases, una de actualización y

otra de maximización, que se repiten en sucesivas iteraciones. En cada iteración se produce una alineación y el modelo es refinado hasta que no se puedan lograr mayores mejoras.

Finalmente, el muestreo Gibbs resuelve el complejo problema del alineamiento múltiple en cuestión de segundos en ausencia de sistemas expertos o información auxiliar derivada de otras fuentes. Al tener un procedimiento de optimización aleatorizado en lugar de una aproximación determinística, es capaz de localizar patrones biológicamente significativo garantizando velocidad y sensibilidad del algoritmo.

En las secciones siguientes se detalla la justificación biológica que hace posible la aplicación del muestreo Gibbs, así como el modelado estadístico de los motivos. Seguidamente se presenta el algoritmo que es base para la implementación del método en estudio y su posterior codificación en el programa GibbsSM. Éste último se utiliza en la parte experimental para comparar su rendimiento con otros programas de detección de motivos disponibles públicamente.

## 3.2 Problema y justificación biológica

El problema de localizar y describir patrones comunes, que se supone están contenidos en un conjunto de secuencias, está regido por un modelo con ciertas características biológicas. Los patrones están, generalmente, especificados por un número relativamente pequeño de residuos sin espacios. Pueden aparecer en diferentes posiciones dentro de las secuencias debido al reordenamiento genómico, como inserciones, eliminaciones y duplicaciones. Y, son descriptos por un modelo probabilístico de las frecuencias de los residuos en cada posición del mismo. Estas características han sido derivadas de establecer los principios de la estructura de proteínas y el conocimiento de las fuentes de variación de los patrones en secuencias [54].

En este problema, dadas *N* secuencias biológicas $S = (S_1, ...., S_N)$, inicialmente desalineadas, se desea localizar un *motivo* de ancho *w* en cada una de ellas. La ubicación de cada zona y el modelo probabilístico que rige su patrón son, a priori, desconocidos, y deberán ser determinados experimentalmente.

La hipótesis de que cada secuencia tiene exactamente un motivo no es realista. Los biólogos han demostrado que existen múltiples motivos en cada secuencia, y en ciertos casos de secuencias poco relacionadas no existe ningún motivo en ellas. Sin embargo, como en el caso de Liu *et al.* 1995 y Lawrence *et al.* 1993, se desarrollaron algoritmos que permiten la detección y optimización simultánea de múltiples motivos y motivos repetitivos.

## 3.3 Modelo estadístico para motivos simples

Dado un conjunto de secuencias que comparten un motivo de ancho *w*, el problema consiste en ubicar el motivo en cada secuencia. De forma que se desea una alineación local múltiple. La alineación múltiple queda definida a través de un vector $A=(a_1,.....,a_N)$ que indica la posición de comienzo del motivo en cada una de las secuencias dadas. Luego, *A* es la incógnita de este problema.

El objetivo es calcular la probabilidad a posteriori de *A* dadas las secuencias $S=(S_1,....,S_N)$, es decir, $P(A|S)$. Según la definición de probabilidad condicional $P(A|S)=P(A,S)/P(S)$. Dado que $P(S)$ representa la probabilidad de observar el conjunto de secuencias *S*, sólo juega el rol de una constante de normalización, entonces la probabilidad a posteriori resulta proporcional a la probabilidad conjunta:

$$P(A|S) \propto P(A,S) \qquad (3.1)$$

No obstante este es un problema complicado debido a que *A* es un dato oculto. Y por lo tanto se deberán imponer restricciones para poder proponer alguna solución. Si se conoce el modelo estadístico de los motivos, ayudaría a resolver el problema, aunque aumenta la dimensionalidad del sistema. Una consecuencia natural de la alineación es que permite definir el modelo estadístico de los motivos y recíprocamente, encontrado los motivos se obtiene el perfil estadístico de la alineación local de las secuencias. Por lo cual se agrega el modelo estadístico del problema: *perfil* y *background*.

Se supone que la ocurrencia de los residuos fuera del motivo, en el *background*, son independientes y siguen un modelo común multinomial de parámetros $q_0=(q_{1,0},....,q_{d,0})$, donde $q_{0,i} \geq 0 \quad \forall i=1...d$ (d=4 para ADN y d=20 para proteínas) y $|q_0|=1$. Al mismo tiempo, las frecuencias de los residuos en cada posición de los motivos son modeladas por productos multinomiales con parámetros, el *perfil*, $\Theta=[q_1,....,q_w]$, donde cada $q_i=(q_{1,i},...,q_{d,i})$ sigue una distribución multinomial. Estos vectores columna $q_i$ son mutuamente independientes y cada uno especifica la probabilidad que un residuo dado pueda ser observado en la posición *i-ésima* del motivo.

Los $q_{i,j} \geq 0$ y $|q_i|=1 \quad \forall i=1..w \quad \forall j=1..d$. De aquí se deduce que se necesitan *(w+1)* vectores columnas *d*-dimensionales para describir los datos.

Debe notarse, sin embargo, que el cálculo de $P(A|S)$ requiere de forma implícita el cálculo de un modelo estadístico $q=(q_0,\Theta)$ para la composición del motivo, el cual a la vez requiere la propia

distribución $P(A|S)$. No debe sorprender entonces la propuesta de una solución iterativa: se propone una alineación inicial $A^{(0)} = (a_1^{(0)},....,a_N^{(0)})$ y a partir del mismo se determina un modelo estadístico para la composición $q^{(0)}$. A partir de $q^{(0)}$ puede obtenerse un $A^{(1)}$ y luego un $q^{(1)}$, a partir de este un $A^{(2)}$, y así sucesivamente. Se verá como plantear una solución a este problema.

Suponiendo una distribución a priori sobre $q$ se integra de manera de evitar un muestreo condicional sobre $q$, en términos de Liu (1994) [38] se *colapsa* $q$. Luego, el cálculo de $P(A|S)$ en términos de $q$ puede expresarse como sigue:

$$P(A|S) \propto P(A,S) = \int P(A,S,q)dq = \int P(A,S|q)\, f(q)\, dq$$

reemplazando el parámetro $q$ por $(q_0, \Theta)$ resulta:

$$P(A|S) \propto P(A,S) = \iint P(S,A|q_0,\Theta)\, f(q_0)\, f(\Theta)\, dq_0 d\Theta \qquad (3.3)$$

Aunque (3.3) no resuelve la mayor dificultad del problema bajo estudio, expresa claramente la relación estrecha y compleja entre los problemas de alineamiento y composición del motivo. En general el cálculo de integrales del tipo (3.3) puede resolverse de forma práctica por métodos de Monte Carlo sujeto a que el número de variables no sea elevado (MacKay 2003) y que $P(A,S|q_0,\Theta)$ tenga una expresión cerrada. En este problema, estas dos condiciones no se cumplen. En (3.3) las variables involucradas son las correspondientes al alineamiento y las correspondientes a la composición del motivo. Mas aún, no es posible una expresión cerrada de $P(A,S|q_0,\Theta)$ aún suponiendo una distribución a priori conveniente.

De la discusión anterior surge que una estrategia para resolver (3.3) mediante métodos de muestreo de Monte Carlo debería considerar una reducción significativa del número de variables. Sea $A_{[-k]}$ el alineamiento obtenido sobre el conjunto de secuencias *S* excluida la secuencia $S_k$. Entonces, puede demostrarse que $P(a_k \mid A_{[-k]}, S) \propto P(A|S)$ (página 959 de Liu 1994 [38]), siendo $a_k$ el inicio del motivo en la *k-ésima* secuencia. Dicho de otro modo, la distribución a posteriori del alineamiento $P(A|S)$ puede muestrearse indirectamente a través de $P(a_k \mid A_{[-k]}, S)$, lo cual es particularmente conveniente porque $P(a_k \mid A_{[-k]}, S)$ es univariada. De esta manera se trata a las $a_j$, $j \neq k$ como

constantes. Consecuentemente la distribución predictiva condicional de $a_k$ es fácil de muestrear por tener todos sus componentes conocidos. Estas consideraciones se reflejan como sigue:

$$P(a_k \mid A_{[-k]}, S) \propto P(S, A_{[-k]}, a_k) \propto P(S_k, a_k \mid A_{[-k]})$$

donde $P(a_k \mid A_{[-k]}, S)$ puede calcularse directamente usando un modelo $\{q_0, \Theta\}$ obtenido sobre el conjunto reducido de secuencias $S_{[-k]}$ con alineamiento conocido $A_{[-k]}$. Esto implica que condicionando sobre los valores $a_1,...,a_{k-1},a_{k+1}...,a_N$ y los datos observados *S*, el valor de $a_k$ puede ser fácilmente actualizado por muestreo sobre su distribución condicional predictiva. Luego $P(a_k \mid A_{[-k]}, S)$ puede muestrearse como sigue:

$$P(a_k = j \mid A_{[-k]}, S) \propto \prod_{i=j}^{j+w-1} \left( \frac{q_{c_k, i-j+1}}{q_{c_k, 0}} \right) \qquad (3.4)$$

donde $c_k$ es un residuo de la secuencia $k-ésima$, $q_{c_k, i-j+1}$ es la probabilidad de aparición del residuo $c_k$ en la posición $i-ésima$ del motivo, y $q_{c_k, 0}$ es la probabilidad de aparición del residuo $c_k$ en el background.

En (3.4) la probabilidad que el segmento de longitud *w* en la secuencia *k-ésima* que comienza en la posición *j* es proporcional a la relación de likelihood de pertenecer al motivo o pertenecer al background. En el algoritmo, que se verá luego, este proceso está referido como *muestreo*. El modelo de probabilidades $\{q_0, \Theta\}$ se calcula tomando los valores de $A_{[-k]}$, la *actualización predictiva*.

De esta forma, se pueden actualizar una a una las posiciones del alineamiento *A* en *N* iteraciones, uno por secuencia. Por construcción los valores de $A = (a_1,.....,a_N)$ son muestreos de la distribución de probabilidad buscada $P(A \mid S)$. Además genera una cadena de Markov con datos ocultos. El proceso se inicia de nuevo hasta que el sistema converja a una probabilidad de transición estacionaria $P(a_1,.....,a_N \mid S) = P(A \mid S)$ que es lo buscado.

A los efectos de calcular (3.3), debe asumirse una distribución *a priori* sobre cada una de las componentes multinomiales de $\{q_0, \Theta\}$ que capture el problema bajo estudio y que sea a su vez computacionalmente conveniente. Una distribución a priori adecuada en este sentido es la distribución

de Dirichlet (MacKay, 2003 [42]). La razón es que la distribución a posteriori de un vector aleatorio $[n_1,...,n_d]$ con distribución multinomial $q$ y distribución a priori de Dirichlet de hiperparámetros $a=(a_1,....,a_d)$ es también una distribución de Dirichlet con hiperparámetros $(a_1+n_1,....,a_d+n_d)$. Para el problema de detección de motivos simples, $n_j$ se interpreta como la cantidad de residuos del tipo *j* en el conjunto de *S* consideradas mientras que los $a_i$ se conocen como *"pseudocounts"*. Mas aún, las componentes del valor esperado o media de $q_j$ es $\frac{(n_j+a_j)}{n+a}$ con $1\le j\le d$, donde $n$ es la cantidad total de residuos y $a$ la suma de todos los $a_j$.

A continuación se desarrolla el algoritmo del muestreo Gibbs, base para la implementación del programa GibbsSM.

## 3.4 Algoritmo

La metodología del algoritmo de muestreo Gibbs realizada en esta tesis está basado en el que Lawrence *et al*. (1993) denominaron *site sampler*. En ella se supone que existe un *único* motivo localizado en cada una de las secuencias dadas. El algoritmo se muestra en la figura 3.1.

Este algoritmo encuentra, en forma automática y sin conocimientos adicionales, una alineación local múltiple de ancho *w*, sin permitir inserciones ni espacios. El patrón que caracteriza los motivos es obtenido por localización del alineamiento que maximice la relación entre la probabilidad correspondiente al motivo con la probabilidad del background.

**Muestreo Gibbs "*site sampler*"**

**Datos:** el conjunto de N secuencias y la longitud w del motivo.

**Metodología:**

Elegir la posición de los motivos en forma aleatoria para cada una de las secuencias, $a_j^{(0)}$ para $j = 1,\ldots, N$, $A^{(0)} = (a_1^{(0)},\ldots,a_N^{(0)})$

Repetir

Seleccionar una secuencia por vez $S_k$ y quitarla

⇒ Calcular $M=\{q_0, \Theta\}$ dados los valores actuales de A, usando todas las secuencias menos la k-ésima. **(actualización)**

⇒ determinar $a_k$ de acuerdo a la distribución predictiva

$$P(a_k \mid a_1^{(t+1)},\ldots,a_{k-1}^{(t+1)}, a_{k+1}^{(t)},\ldots, a_N^{(t)}, S^{(-k)})$$

actualizar la alineación tomando $a_k^{(t+1)} = a_k$ . **(muestreo)**

Resulta una nueva aproximación $a_j^{(t+1)}$ a partir de $a_j^{(t)}$ para $j = 1\ldots N$

hasta que converja

**Resultados**: $q_0$, $\Theta$, A

Fig. 3.1. Algoritmo del muestreo Gibbs

Se supone que se dispone de un conjunto de *N* secuencias y se busca dentro de cada una de ellas segmentos similares de ancho *w* especificado. El algoritmo trabaja con dos estructuras de datos que se actualizan en cada iteración. La primera es la descripción del *perfil*, en forma de un modelo probabilístico de residuos en cada posición $i = 1..w$, el vector $\Theta = [q_1,....,q_w]$, y las frecuencias de *background* $q_0 = (q_{1,0},....,q_{d,0})$. La segunda es el arreglo de posiciones $A = \{a_k, k = 1..N\}$.

El algoritmo se inicia determinando en forma aleatoria las posiciones de los motivos en las $N$ secuencias, obteniendo un vector $A^{(0)} = (a_1^{(0)},....,a_N^{(0)})$ .

En el paso iterativo, una de las $N$ secuencias es elegida en forma aleatoria o en un orden específico, con el objetivo de encontrar un nuevo sitio en ella que ajuste mejor el modelo que el anterior.

Se calculan las matrices de probabilidades del perfil, $\{q_0, \Theta\}$ tomando los valores actuales de *A* en todas las secuencias menos la quitada. A esta fase del algoritmo se la denomina de *actualización predictiva.*

Luego, cada segmento de longitud *w* de la secuencia quitada, la *k-ésima,* es considerado como un posible motivo. A cada uno de estos segmentos se le asigna un peso dado por la relación de probabilidades de pertenecer al perfil o pertenecer al background.

$$A_j = \prod_{i=j}^{j+w-1} \left( \frac{q_{c_k,i-j+1}}{q_{c_k,0}} \right)$$

$A_j$ es el peso del segmento que comienza en la posición *j-ésima* de la secuencia *k* de longitud $L$ , con ancho *w*. De acuerdo a estos pesos uno de ellos es seleccionado en forma aleatoria. La posición de inicio de este segmento seleccionado es el nuevo sitio $a_k$ en la secuencia *k-ésima*. A esta fase del algoritmo se la denomina de *muestreo.*

Este simple proceso iterativo constituye el algoritmo básico. La idea central es que cuanto más precisa es la descripción del perfil construido en el primer paso, más precisa va a ser la determinación de los $a_k$ en el segundo paso, y viceversa. Tomando posiciones $a_k$ al azar, la descripción $\Theta$ del perfil no tenderá a favorecer a un segmento en particular. Una vez que un $a_k$ correcto ha sido seleccionado, los $\Theta$ empiezan a reflejar, aunque imperfectamente, un perfil extendido en las demás

secuencias. Este proceso tiende a coleccionar nuevos $a_k$ correctos, los cuales irán mejorando el desarrollo del perfil.

Este algoritmo es de hecho un muestreo predictivo donde la probabilidad de determinar el $a_k$ en un tiempo iterativo *t* depende de los $a_k$ anteriores y de las *N-1* secuencias restantes.

Estos pasos se repiten hasta converger a un valor que maximice el alineamiento o cuando se haya iterado una cantidad de veces especificada (como parámetro del programa).

El problema de identificar estos sitios a partir de un conjunto de secuencias desalineadas se puede resumir como el problema de encontrar valores del vector *A* que maximice las *similaridades* entre los motivos. Esta similaridad entre segmentos de residuos puede ser representada por la información contenida o distancia de Kullback-Leibler entre el motivo y el background, que puede expresarse como sigue:

$$I = \sum_{i=1}^{w} \sum_{j=1}^{d} h(r_{i,j}) \log \frac{q_{i,j}}{q_{o,j}}$$

Donde $h(r_{i,j})$ es la cantidad de residuos de tipo *j* observados en la posición *i-ésima* del motivo.

Este valor es máximo si el motivo está bien conservado y difiere considerablemente de la distribución de background.

Storno y Hartzell (1989) propusieron un primer método, voraz pero efectivo, para solucionar el problema de localización de motivos. En él escogen el motivo más probable según la distribución de frecuencias de los residuos, el que maximiza $A_j$. En 1993, Lawrence *et al.* [32] le incorporan procesos de optimización heurística logrando un algoritmo más rápido y eficiente, evitando caer en máximos locales, el muestreo Gibbs analizado en esta tesis.

### 3.4.1 Implementación

Se implementó el algoritmo recién descripto utilizando la siguiente notación:

**Notación**

- $N$ : cantidad de secuencias
- $S_1,...,S_N$ : secuencias dato
- $w$: ancho del motivo a ser encontrado en las secuencias
- *n_alphabet* : cantidad de residuos en el alfabeto. 4 para secuencias de ADN y 20 para secuencias de proteínas.
- $c_{ij}$ : cantidad observada de residuos $j$ en la posición $i$ del motivo, *con* $j=1...n\_alphabet$ e $i=1...w$ .
- $c_{oj}$: cantidad observada de residuos $j$ fuera de los motivos (el background) *con* $j=1...n\_alphabet$
- $q_{ij}$ : frecuencia del residuo $j$ ocurrida en la posición $i$ del motivo, con $i=1...w$.
- $p_j$ : frecuencia del residuo $j$ ocurrida en el background
- $a_k$ : vector que contiene las posiciones de inicio del motivo dentro de cada secuencia, con $k=1..N$ . Éstos son los parámetros ocultos del problema.
- $b_j$ : pseudocounts para cada residuo, necesarios para eliminar los problemas con las cantidades nulas en la estadística bayesiana.
- $d_j$ : pseudocounts del background.
- B: suma de los pseudocounts de los residuos. $\sum_{j=1,n\_alphabet} b_j$
- Q: suma de los pseudocounts del background. $\sum_{j=1,n\_alphabet} d_j$

## Datos

- ✓ N secuencias en formato FASTA (de proteínas o ADN)
- ✓ El ancho *w* del motivo

## Salidas

- ✓ El perfil del motivo (modelo probabilístico)
- ✓ Una tabla que contiene por cada secuencia: nombre de la secuencia – posición de inicio del motivo- secuencia de residuos del motivo- posición de terminación del motivo – los 10 residuos anteriores y posteriores al motivo.

## Metodología

#### Inicialización

Una vez que las secuencias son conocidas, las cantidades de cada tipo de residuo son calculadas. Inicialmente $c_{oj}$ contiene la cantidad total del residuo j dentro de todas las secuencias, y $c_{ij}$ es inicializado en 0 para todos los valores de i.

Luego de seleccionar en forma aleatoria las posiciones de comienzo de los motivos dentro de las distintas secuencia, el vector A, se vuelve a calcular los $c_{ij}$ y los $c_{oj}$, y con ellos los $q_{ij}$ y los $p_j$ **.** De igual forma que el modelado estadístico visto en el Capítulo 2, se sugiere que los $c_{ij}$ y los $c_{oj}$ sean suplementados con valores de pseudocounts. Los $q_{ij}$ y los $p_j$ son calculados usando las siguientes ecuaciones.

$$q_{ij} = \frac{c_{ij} + b_j}{N - 1 + B}$$ **ecuación 1** : frecuencias de residuos en el motivo

$$p_j = \frac{c_{0,j} + d_j}{total + Q}$$ **ecuación 2** : frecuencias de residuos en el background

*total* es la cantidad total de residuos en las *N* secuencias sin contabilizar los residuos de los motivos.

Se puede elegir $q_{ij}$ simplemente proporcional a $c_{ij}$ pero esto implica una probabilidad 0 para cualquier residuo que no sea observado. Una solución común, cuando son considerados los modelos multinomiales, es la distribución de Dirichlet. Pero si existe un conocimiento previo de la densidad de distribución de los residuos dentro de la alineación, estos pseudocounts son valores dependientes de dicho conocimiento. Si esta información no es conocida, se puede pensar proporcional a la cantidad de cada residuo en las *N* secuencias dadas.

**Inicialización de los pseudocounts**

```
//pseudo[j]: pseudocounts del residuo j / pseudo0 = suma de pseudocounts

        PSEUDO_RATIO = 0.0001; // Proporcionalidad de pseudocounts
        pseudo0 = 0.0;
        for (int n = 0; n < nalpha; n++) {
           pseudo[n] = PSEUDO_RATIO * tot[n];
           pseudo0 += pseudo[n];
        }

//  tot[n] es la cantidad total del residuo n-ésimo del alfabeto.
```

Fig.3.2. Algoritmo de inicialización de los pseudocounts

**Iteración**

El algoritmo itera sobre 2 pasos, el predictivo y el de muestreo.

El primer paso , la actualización *predictiva*, selecciona una de las N secuencias, la secuencia *z*, en forma aleatoria o en forma secuencial (cualquiera de las 2 formas). La secuencia z es quitada del modelo. Luego se calcula nuevamente los $c_{ij}$ y los $c_{oj}$ para actualizar los $q_{ij}$ y los $p_j$ desde las posiciones $A_{[-k]}$ actuales, excluyendo la secuencia *k*.

El segundo paso, el de *muestreo*, todo segmento *x* de ancho *w* dentro de la secuencia *k* es considerado una posible instancia del motivo. Se ventanea la secuencia *k* en segmentos de longitud *w*. Para cada uno de esos segmentos *x*, se calcula la probabilidad $Q_x$ de ser generado por el perfil $q_{ij}$, y la probabilidad $P_x$ de ser generado por la probabilidad de background $p_j$. Esto es, evaluar cada segmento *x* para saber si puede pertenecer o no al perfil.

$$Q_x = \prod_{i=1}^{w} q_{i,r_i}$$ **ecuación 3**: es la probabilidad de que se generara el segmento x si el mismo perteneciera al motivo

$$P_x = \prod_{i=1}^{w} p_{r_i}$$ **ecuación 4**: es la probabilidad de que se generara el segmento x si el mismo perteneciera al background

$$A_x = \frac{Q_x}{P_x}$$ **ecuación 5**: peso del segmento x

Se calcula el valor $A_x$ que es asignado como peso a cada segmento *x*. Después de la normalización, $A_x$ da la probabilidad que el motivo en la secuencia *k* pertenezca al sitio *x*. Con estos pesos relacionados a los diferentes segmentos *x*, uno de ellos es seleccionado en forma aleatoria. De este modo, los *x* con mayor peso serán más probables de ser elegidos que aquellos que tengan menores valores. Pero como en todo proceso estocástico, no se garantiza escoger el sitio *x* con el peso más alto. A la posición de inicio de ese segmento seleccionado se la denomina $a_k$ , y pasa a ser el inicio del motivo dentro de la secuencia *k-ésima*.

**Obtención de los resultados**

Una vez que los pasos de iteración se realizaron para todas las secuencias, se obtiene un alineamiento $A = \{a_k, k = 1..N\}$. Para este alineamiento, se calcula F.

$$F = \sum_{i=1}^{w} \sum_{j=1}^{n\_alphabet} c_{i,j} \log \frac{q_{i,j}}{p_j}$$ **ecuación 6:** log-likelihood de la alineación

El objetivo es maximizar F, para ello se realizan varias corridas del algoritmo comparando los valores de F obtenidos en cada una y guardando los perfiles que la hagan máxima.

**Algoritmo para hallar el alineamiento más probable**

```
GlobalMaxAlineaProb=0
Repetir para  n = 1 to nrun
{       Inicializar el alineamiento en forma aleatoria
        localMaxAlineaProb = 0;
        mientras (no se encuentre un máximo local & iter < iter_max)
        {  realizar para cada secuencia
              {    quitar la secuencia del alineamiento
                   realizar una actualización Predictiva
                   realizar el muestreo con la secuencia quitada
              }
           calcular AlineaProb
           Si (AlineaProb > localMaxAlineaProb)
              {  localMaxAlineaProb = AlineaProb;
                 máximo local = verdadero;
              }
              iter++;
        }
        Si (localMaxAlineaProb == GlobalMaxAlineaProb)
              Salir ( "el máximo se encontró dos veces ")
         Sino Si(localMaxAlineaProb > GlobalMaxAlineaProb)
                 GlobalMaxAlineaProb = localMaxAlineaProb
}
```

Fig.3.3. Algoritmo para hallar el alineamiento más probable

**Estructuras de datos**

En el algoritmo se distinguen 2 estructuras de datos que se van actualizando en cada iteración:

1) la que **describe el perfil.** Consiste en 2 arreglos, una matriz *q* de dimensión *n_alphabet* filas por *w* columnas, donde *w* es el parámetro que representa el ancho del motivo a buscar, y *n_alphabet* es la cantidad de símbolos del alfabeto de las secuencias (20 si son proteínas o 4 si son nucleótidos) que contiene las frecuencias de los residuos en cada posición del patrón; en el programa representado por *site_freq[ ][ ].* Un vector *p* de dimensión *n_alphabet* que contiene las frecuencias del background, es decir, la frecuencia de los residuos que no pertenecen a los motivos; en el programa es representado por *nonsite_freq[ ]* .

2) la que contiene las **posiciones de comienzo del motivo en cada secuencia**. Consiste en un vector denominado a , donde cada $a_k$ , con k= 1 a N, indica el comienzo del motivo en cada una de las N secuencias; en el programa representado por *pos[ ]*.

### 3.4.2 Inconveniente

Un inconveniente de este algoritmo es que puede caer en un máximo local sin encontrar el óptimo. Para ello en Lawrence *et al*. (1993) [32] se propone realizar un corrimiento del patrón a derecha e izquierda para ver si el valor de las probabilidades varían. La idea es insertar un nuevo paso en el algoritmo denominado *shifted*. Luego de cada iteración, se compara el *A* con otros alineamientos corridos a izquierda y derecha en la misma cantidad de posiciones, generalmente pequeña, en todas las secuencias (en el programa gibbs9_95 [GIBBS_SOFT] se realiza el corrimiento de sólo 2 residuos a izquierda y a derecha). Las relaciones probabilísticas son calculadas para estas 2 posibilidades, y una es seleccionada en proporción a sus pesos de probabilidades.

### 3.4.3 Complejidad

Sea $N$ la cantidad de secuencias de un alfabeto de $d$ letras, $L^m$ la longitud media de las secuencias y $w$ el ancho del patrón. Entonces, si el programa trabaja con las secuencias desde disco, como suelen hacerlo la mayoría de los programas en Bioinformática, el requerimiento de memoria no es mucho: una secuencia y 3 arreglos, el alineamiento (vector de $N$ elementos) y el perfil (una matriz de $d$ filas y $w$ columnas y un vector de $d$ elementos). Si el programa almacena las $N$ secuencias en memoria, el requerimiento es $NL^m$ y los 3 arreglos (alineamiento y perfil).

La complejidad temporal es más difícil de calcular. En el primer paso los cálculos se basan en sumas, para hallar los $q_{ij}$ se realizan $w(N-1)$ sumas; para los $p_j$ se realizan $(N-1)L^m$ sumas. Este es el peor caso de diseño, pues se puede tener los cálculos de los $p_j$ realizados al principio del programa y luego en cada iteración restarle los residuos que figuran en los motivos de cada secuencia, en este caso habría $w(N-1)$ restas. En el segundo paso el tiempo consumido es aproximadamente $wL^m$ multiplicaciones, donde $L^m$ es la longitud media de la secuencia $k$ . Dado que son N secuencias resulta la complejidad en $N\,L^m\,w$.

Por consiguiente, el número total de multiplicaciones necesarias para ejecutar el muestreo Gibbs es aproximadamente $N\,L^m\,w$ . Como el valor de $L^m$ es generalmente mayor a $N$ y a $w$ entonces el orden de complejidad es $\simeq O(L^m)$ . Concluyendo que el método tiene complejidad lineal.

## 3.5 Problemas abiertos sobre muestreo Gibbs

Frecuentemente a los biólogos les interesa saber cuales son los motivos en las secuencias sin especificar el ancho del mismo. Sin embargo, el método de determinar motivos del algoritmo antes descripto requiere que el ancho del motivo sea dato del problema, y permanezca fijo durante la ejecución del programa. Por este hecho, se sugiere ampliar este algoritmo incorporándole un rango de valores de $w$, de tal manera de determinar el tamaño del modelo más probable. Así como también, redefinir el método para permitir espacios dentro del motivo y/o localizar múltiples motivos de diferentes tipos. En este sentido, suele ocurrir que algunas secuencias del conjunto no estén relacionada con las demás y por consiguiente no tenga motivos en común. En este caso el algoritmo debe determinar esta situación y no considerar ningún motivo en ella. La última observación está relacionada con los pseudocounts. Se plantea en el algoritmo propuesto que los pseudocounts son

dependientes de la cantidad de residuos de cada tipo en todas las secuencias dadas, la sugerencia en este caso es permitir introducir un vector de pesos de pseudo-counts que resulte del conocimiento a priori que se disponga de las propiedades biológicas de las secuencias en juego.

## 3.6 Alternativas al problema de detección de motivos

### 3.6.1 MEME

En esta sección se describe brevemente los fundamentos del programa MEME (Multiple EM for Motif Elicitation) desarrollado por Timothy Bailey y Charles Elkan[5] en 1994 para identificar automáticamente motivos conservados en un conjunto de secuencias de ADN o de proteínas. MEME[MEME] requiere que se precise el ancho del motivo, y no considera ni inserciones ni blancos en el motivo. A diferencia del muestreo Gibbs, MEME resuelve el problema asumiendo un modelo probabílistico usando EM (ver sección 2.7) para la estimación MLE de los parámetros del modelo. En este caso no se considera la estimación a priori de la localización de los motivos ni de la composición de los mismos. Este programa, además, determina el valor de significancia estadística de cada uno de los motivos hallados en cada secuencia. Este valor es calculado utilizando la estadística enunciada por Karlin y Altschul (1990) [30]. Esta información adicional permite al biólogo que se forme una opinión crítica de la validez de los resultados. Incluso se dispone de un servidor web de fácil utilización en http://meme.sdsc.edu/meme/website/.

A continuación se presenta el algoritmo que implementa el método EM a la localización de motivos en secuencias biológicas. Cabe puntualizar que en la aplicación del método EM a la búsqueda de motivos, la matriz $Z$ de posición de los motivos y la matriz $Q$ de composición de los mismos están determinadas considerando todas las posibilidades con todas las secuencias.

Sean *m* secuencias $X$ cuyo motivo se supone de ancho *w*. Sea $Z$ una matriz cuyos elementos $Z_{ij}$ representan la probabilidad que el motivo comience en la posición $j$ de la secuencia $i$, y $Q$ otra matriz con $w+1$ columnas y $4$ filas (por suponer secuencias de ADN), cuyos elementos $q_{ck}$ representan la probabilidad de aparecer el residuo $c$ en la columna $k$ del motivo; en la columna $0$ están las probabilidades del background. En la aproximación EM se supone tener valores iniciales de $Q$. A partir de $Q$ se estiman los valores de $Z$ (paso E) y con dichos valores se re-estiman los valores de $Q$

(paso M). Luego se repiten los pasos E y M hasta que los cambios en $Q$ sean < $\varepsilon$. En la figura 3.4 se describe el algoritmo de este método.

Paso E

Sea $X_i$ la *i-ésima* secuencia, además sea $Z_{ij} = 1$ si el motivo de la secuencia *i* comienza en la posición *j*. Los $Z_{ij}$ se calculan aplicando Bayes a $P(Z_{ij} = 1 \mid X_i, Q^{(t)}) \equiv P(y \mid x, q^{(t)})$,

obteniendo $$Z_{ij} = \frac{P(X_i \mid Z_{ij} = 1, Q^{(t)}) P(Z_{ij} = 1)}{\sum_{k=1}^{L-w+1} P(X_i \mid Z_{ik} = 1, Q^{(t)}) P(Z_{ik} = 1)}$$

Si se asume que toda posición es igualemente probable para el comienzo del motivo, resulta

$$Z_{ij} = \frac{P(X_i \mid Z_{ij} = 1, Q^{(t)})}{\sum_{k=1}^{L-w+1} P(X_i \mid Z_{ik} = 1, Q^{(t)})}$$

La probabilidad de la secuencia *i-ésima* suponiendo que el motivo comienza en el residuo *j-ésimo* es

$$P(X_i \mid Z_{ij} = 1, Q) = \prod_{k=1}^{j-1} q_{c_k,0} \prod_{k=j}^{j+w-1} q_{c_k,k-j+1} \prod_{k=j+w}^{L} q_{c_k,0}$$

donde $q_{c_k,k}$ es la probabilidad del residuo $c_k$ en la posición *k* de la secuencia; cuando *k=0* se refiere al background.

---

Paso M

Se debe recalcular los valores de la matriz $Q$ a partir de los *Z* estimados en el paso anterior.

Sea $q_{c,k}^{(t+1)}$ la probabilidad del residuo *c* en la posición *k* en una iteración *(t+1)*. Y suponiendo una distribución Dirichlet sobre éstos parámetros, resulta

$$q_{c,k}^{(t+1)} = \frac{(n_{c,k} + a_{c,k})}{N + a}$$

Donde $n_{c,k} = \sum_i \sum_{\{j \mid X_{i,j+k-1} = c\}} Z_{ij}$ es la suma de todos los valores de *Z* en todas las secuencias cuya posición k-ésima del motivo comience con el residuo *c*.

*N*= total de residuos *c* en todas las secuencias – los residuos *c* que aparezcan en los motivos.

Los $a$ están referidos a los pseudocounts.

Luego se calcula el likelihood de los datos dado el modelo $\prod_i \sum_{k=1}^{L-w+1} P(X_i \mid Z_{ij} = 1, Q)$

Fig.3.4. Búsqueda de motivos aplicando EM

### 3.6.2 GIBBS

En esta sección se describe brevemente los fundamentos del programa *Gibbs Motif Sampler* [GIBBS] , de ahora en más GIBBS. Fue desarrollado por Eric C. Rouchka y William Thompson dentro del programa de Bioinformática del Laboratorio de Wadsworth Center. Este programa está basado en el trabajo de Lawrence *et al.* (1993) para identificar automáticamente motivos conservados en un conjunto de secuencias de ADN o de proteínas. Este programa implementa el método de muestreo Gibbs como fue descripto en esta tesis. Sin embargo, a diferencia del muestreo Gibbs implementado en esta tesis, este programa permite determinar más de un motivo por secuencia, además más de un tipo distinto de motivo en una misma secuencia, y también admite la fragmentación de los motivos. La fragmentación se refiere a considerar al motivo en columnas no consecutivas de la alineación, permitiendo una separación del motivo en varios segmentos no consecutivos. Este programa está disponible a través de un servidor web en la página http://bayesweb.wadsworth.org/gibbs/gibbs.html[4], el cual permite analizar las secuencias que se le remiten y devuelve vía mail el resultado de los motivos con el perfil respectivo. La versión web del GIBBS opera en 3 modos básicos: *site sampling*, *motif sampling* y *recursive sampling.* En *site sampling* [32], se supone que cada secuencia contiene un sitio de cada tipo de motivo. En *motif sampling* [39][46], el usuario provee una estimación de la cantidad total de motivos para cada tipo que puede haber en cada secuencia. Con el muestreo *recursive* [40], el usuario debe introducir un límite máximo de la cantidad de sitios por secuencia. Las alternativas *motif sampling* y *recursive sampling* suponen que cada secuencia puede contener varios motivos de cada tipo o ninguno. De aquí se deduce que algunas de las secuencias analizadas podrían no considerarse a la hora de realizar la alineación.

En el Capítulo siguiente, Parte Experimental, se utilizan estos dos programas, MEME y GIBBS, a los efectos de validar los resultados arrojados por GibbsSM en su aplicación a la detección de motivos en secuencias de proteínas del complejo de iniciación en los organismos protistas *Trypanosoma Cruzi* dentro del Proyecto Serra *et al.* 2003.

[4] Este programa fue modificado a partir del 21 de julio 2005, a raíz de un intercambio de correspondencia entre William Thompson ( BrownUniversityCenter for Computational Molecular Biology) y la autora de esta tesis.

# Capitulo 4: Parte experimental

*"En Bioinformática nunca debe confiarse ciegamente en una computadora, sólo un experto humano puede valorar razonablemente la validez biológica de los resultados."*
*José Valverde* (EMBnet/CNB-España)

## 4.1 El programa *GibbsSM*

A los efectos de la aplicación del muestreo Gibbs para la detección de motivos en alineaciones múltiples, se ha desarrollado un programa experimental denominado GibbsSM. Este programa localiza motivos simples de ancho prefijado en secuencias biológicas de proteínas o de ADN, sin considerar ni inserciones ni espacios. Este programa se basó en la metodología de muestreo Gibbs descripta en capítulos anteriores. El programa GibbsSM está escrito en Lenguaje Java y se encuentra disponible en la página web http://www.eie.fceia.unr.edu.ar/~bioinfo/gibbs.

Se eligió el lenguaje Java debido a que produce software portable, es decir, independiente de la arquitectura de la PC, y funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible. Además, al ser un lenguaje orientado a objetos, permitió realizar un diseño compacto y legible del programa; a su vez, las librerías gráficas de Java permitieron realizar la interfaz del GibbsSM, mostrada en la fig. 4.2, con bastante celeridad. Por otra parte se dispone en Internet de una biblioteca de algoritmos y clases en lenguaje Java de código abierto denominada Weka[65][66], *http://www.cs.waikato.ac.nz/ml/weka/*, la cual fue consultada. La licencia de Weka es GPL[5], lo que significa que es de libre distribución y difusión. Esta biblioteca fue desarrollada por la Universidad de Waikato, Nueva Zelanda, y contiene un conjunto de algoritmos de machine learning, entre ellos herramientas de pre-procesamiento, de clasificación, de regresión, de clustering, y asociación de datos. Weka está diseñado como herramienta orientada a la extensibilidad por lo cual es posible añadir nuevas funcionalidades.

[5] GNU Public License.http://www.gnu.org/copyleft/gpl.html

El GibbsSM se compactó como archivo .jar para colocarlo en la página web *www.eie.fceia.unr.edu.ar/~bioinfo/gibbs* con la intención que tanto el código fuente como la aplicación de este software estén disponibles en forma pública. El archivo *gibbsSM.jar* se puede bajar de la página y ejecutar haciendo un doble clic en la aplicación. Hay que tener en cuenta que las aplicaciones Java no corren aisladas por sí solas. De hecho las aplicaciones convencionales que se utilizan diariamente requieren de diversos paquetes y bibliotecas de funciones que en algunas ocasiones ya se encuentran en Windows pero en otras deben ser obtenidas desde Internet en forma libre y gratuita. Tal es el caso de las aplicaciones creadas con Java. Una aplicación Java necesita de la Máquina Virtual Java (que la interpreta) y de una biblioteca de clases (a la que recurre) para poder ejecutarse; ambas se encuentran en el paquete Java Runtime Environment (JRE). Para ejecutar las aplicaciones Java que se encuentran en *.jar es necesario disponer del JRE correspondiente al sistema operativo que incluye parámetros que automáticamente modifican la plataforma para que ejecute los *.jar.

### 4.1.1 Arquitectura

Para ilustrar cómo fue pensado el programa GibbsSM se realizó un esquema con los diferentes subsistemas que componen la aplicación y sus interrelaciones, el cual se muestra en fig.4.1. En el mismo se distinguen claramente cuatro módulos: *Entrada*, *Control*, *Método* y *Vista* de resultados.

El módulo *Entrada* constituye la interfaz con el usuario. En la misma se adquieren los datos necesarios para aplicar el método de muestreo Gibbs: el archivo que contiene las *n secuencias*, el *ancho del motivo*, y los parámetros opcionales. Estos últimos se refieren a: la cantidad de veces que se desea realizar el muestreo Gibbs sobre el conjunto de secuencias (*nrun*), la cantidad máxima de iteraciones para hallar un óptimo local (*iter max*), el valor inicial de la semilla para la función de generación de números aleatorios (*random seed*), el valor de proporcionalidad para los pseudocounts (*pseudo ratio*), el valor del *error* para acotar la diferencia entre el mejor alineamiento encontrado en la etapa *iter* y el alineamiento óptimo encontrado en cada *nrun*, y la marca de *shuffle*, que indica si se debe realizar un cambio aleatorio en el ordenamiento de las secuencias.

Una vez que la información ingresa al sistema es tomada por el módulo de *Control*, el cual se encarga de chequear si los datos ingresados son correctos. En caso de existir algún inconveniente de consistencia se emite el respectivo mensaje de error. Los tipos de errores contemplados se detallan en la siguiente sección *Fuentes de errores comunes en el uso del programa*. En este módulo también se crea el archivo de salida, cuyo nombre es el mismo del archivo que contiene las secuencias aunque la extensión se cambia por *gibbs*.

Si no se detectaron errores, el sistema continúa al módulo de *Método*. En este módulo se realiza la localización de los motivos de las secuencias dadas como información, según la metodología de muestreo Gibbs descripta en los capítulos anteriores. Dentro de este subsistema se encuentran la

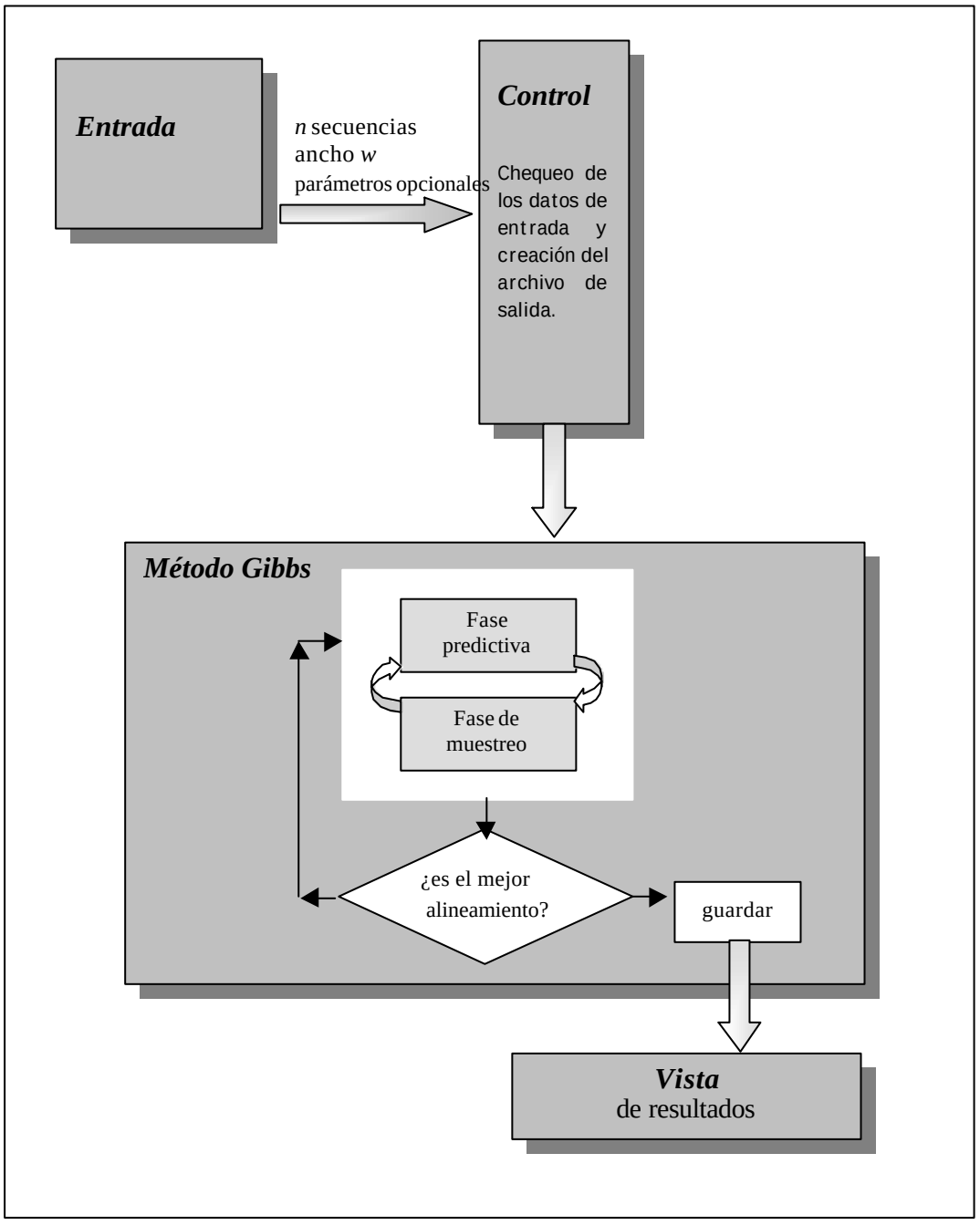


Fig.4.1. Esquema de funcionamiento del GibbsSM

fase predictiva y la de muestreo que son la base de la metodología. Una vez finalizado este proceso se decide si es la mejor alineación o no de acuerdo al valor del log-likelihood dado por la ecuación 6 del Capitulo 3. Si el valor del log-likelihood de la alineación recién obtenido no es mejor que el que está guardado, se descarta y se vuelve a repetir el proceso.

Una vez que el método se ha repetido tantas veces como fue fijado por el usuario en el módulo de Entrada, se pasa al módulo de ***Vista*** para establecer los resultados del sistema. En este módulo se recoge la información guardada de las posiciones de comienzo del motivo en cada una de las secuencias y se organiza la salida de tal manera de encolumnar los motivos en las diferentes secuencias. Conjuntamente se muestra un máximo de 10 residuos anteriores y posteriores al motivo en cada secuencia. Además de los motivos se presenta el perfil que describe el motivo. Este perfil consiste de 2 arreglos: una matriz de dimensión *n_alphabet* filas por *w* columnas que contiene las probabilidades de los residuos en cada posición del patrón y un vector de dimensión *n_alphabet* que contiene las probabilidades de background. Cabe recordar que *w* es dato de entrada y representa el ancho del motivo a buscar, y *n_alphabet* es la cantidad de símbolos del alfabeto de las secuencias: 20 si son proteínas o 4 si son nucleótidos.

### 4.1.2 La interfaz

La interfaz con el usuario se diseñó de forma simple para que el programa pueda ser utilizado por personas no familiarizada con los ambientes de programación. Se utilizó el inglés por ser el idioma mayoritariamente empleado en la comunidad científica que trabaja en el área de la Bioinformática. La interfaz se muestra en la fig. 4.2.

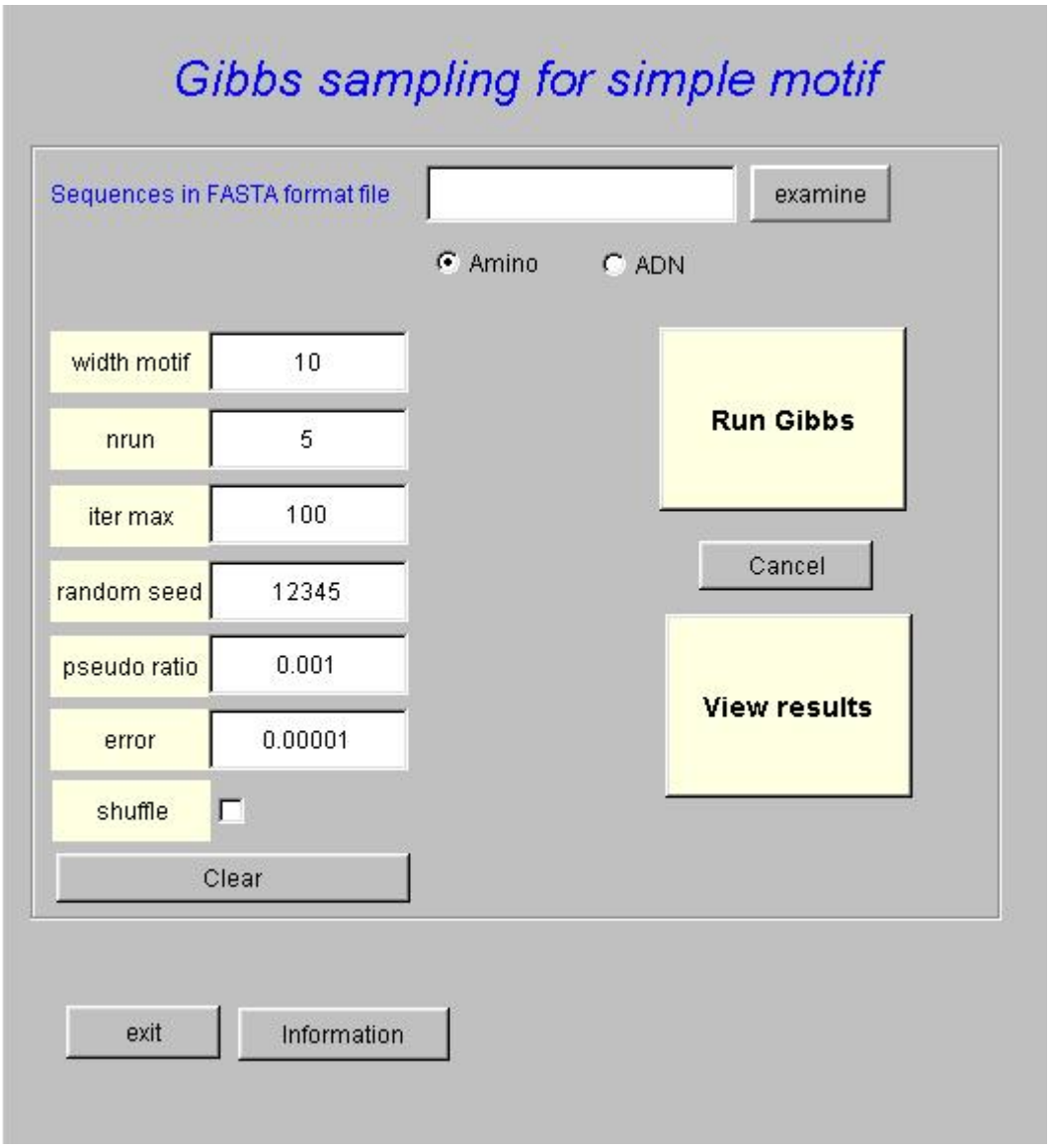


Fig. 4.2. Interfaz del programa GibbsSM

Para ejecutar el programa son necesarios 3 datos fundamentales:

1) El archivo que contiene el conjunto de secuencias en formato FASTA. Se debe usar el botón *examine* que permite ubicarlo. Es de notar que las secuencias en el archivo deben estar separadas por un renglón en blanco.
2) La longitud del motivo (*motif width*). Debe ser un número entero.
3) El tipo de secuencias: aminoácidos o ADN. Para seleccionarlo se debe pulsar el botón correspondiente. Por default el programa considera que son aminoácidos.

Las demás opciones permiten el control de los parámetros de ejecución del programa:

4) *nrun*, es el número de veces que se desea realizar el muestreo Gibbs sobre el conjunto de secuencias, debe ser al menos igual a uno. Por defecto es 5.
5) *iter max*, la cantidad máxima de iteraciones para hallar un óptimo local. Por defecto es 100.
6) *random seed*, valor inicial de la semilla para la función de generación de números de aleatorios. Por defecto es 12345.
7) *pseudo ratio*, valor de proporcionalidad para los pseudocounts. Por defecto es 0.001.
8) *error*, para acotar la diferencia entre el mejor alineamiento encontrado en la etapa *iter* y el alineamiento óptimo encontrado en cada *nrun*. Por defecto es 0.00001.
9) *shuffle*, realiza un cambio aleatorio en el orden de las secuencias dadas. Por defecto no está habilitado.

## Botones en la interfaz

*Clear*, permite borrar el nombre del archivo y pone los diferentes parámetros opcionales en sus valores por defecto.
*Run Gibbs*, permite iniciar la ejecución del programa de muestreo Gibbs.
*Cancel*, permite cancelar la ejecución.
*View results*, muestra los resultados del archivo.*gibbs* en una ventana de la interfaz.

### 4.1.3 Fuentes de errores comunes en el uso del programa

Los errores más comunes al utilizar este programa son:

1- **Los caracteres que componen las secuencias no son admitidos:** es el error más común que se puede presentar. Se presenta cuando los caracteres que componen las secuencias contenidas en el archivo no sean reconocidos como aminoácidos o nucleótidos, dependiendo en cada caso del alfabeto utilizado. Un mensaje de error muestra que símbolo no pertenece a dicho alfabeto y la ejecución del programa se detiene. Para solucionar este problema se debe editar el archivo de secuencias y asegurarse que todos los residuos de las secuencias pertenezcan al alfabeto seleccionado. Por otra parte las secuencias deben estar en formato FASTA, y entre ellas debe haber un renglón en blanco.

2- **No colocar el archivo que contiene las secuencias:** cuando no se coloca ningún nombre de archivo en el espacio reservado en *examine*, y se ejecuta el programa. El programa no se ejecuta.

3- **La longitud del motivo es mayor que la secuencia**: puede suceder que el ancho del motivo a buscar sea mayor que la longitud de alguna secuencia. En este caso se muestra un mensaje de error y la ejecución del programa se detiene. La solución es colocar un ancho de motivo coherente con el conjunto de secuencias.

En la sección siguiente se analizarán grupos de secuencias biológicas relacionadas con el Proyecto *Trypanosoma cruzi* obtenidas en el IBR [56].

## 4.2 Detección de motivos en el *Trypanosoma cruzi*

Para testear el funcionamiento del programa GibbsSM se prepararon cuatro grupos de secuencias de proteínas. Estos grupos analizados fueron obtenidos de la base de datos Pfam [PFAM] bajo la supervisión del Dr. Esteban Serra del Instituto de Biología Molecular y Celular de Rosario de la Facultad de Ciencias Bioquímicas y Farmacéuticas de la Universidad Nacional de Rosario, República Argentina, en el contexto del Proyecto *Caracterización de factores basales de transcripción en parásitos protozoarios*, CONICET (Serra *et al*. 2003-2004 [56]). Este Proyecto se realizó en colaboración del grupo de Parasitología Molecular del Instituto de Biología Molecular y Celular de Rosario. En dicho laboratorio se desarrollan proyectos de biología experimental sobre distintos modelos parasitarios, con particular énfasis en el protozoo *Trypanosoma cruzi*, responsable de la enfermedad de Chagas. El proyecto genoma de este parásito se ha desarrollado durante los últimos años de modo tal que un "draft" del mismo será publicado próximamente. *Trypanosoma cruzi*, al igual que otros parásitos protozoos, presenta una alta divergencia a nivel de secuencia con los organismos eucariotas[6] más estudiados, por lo que el análisis de las secuencias genómicas mediante las metodologías clásicas de alineamiento global o local son insuficientes para la identificación de genes de interés. Esta limitación puede ser sobrellevada mediante el uso de programas de definición y búsqueda de dominios, como el desarrollado en la presente Tesis.

La totalidad de las pruebas fueron ejecutadas en una PC con procesador tipo Pentium II 450MHz con 128MB de RAM y disco rígido de 10GB.

[6] cuyas células tienen núcleo

### 4.2.1 Secuencias analizadas

Se seleccionaron cuatro grupos de secuencias [7] para comparar el rendimiento del programa GibbsSM con otros programas de detección de motivos disponibles públicamente. Para el presente estudio los grupos corresponden a cuatro familias de proteínas. Estos grupos fueron seleccionados teniendo en cuenta su importancia para los proyectos que se llevan a cabo en el Laboratorio del IBR. Se consideró además la experiencia que se contaba con respecto a los mismos en el laboratorio de modo de facilitar cualquier análisis posterior de los resultados. Por otra parte, los grupos se eligieron de tal manera que representaran distintas situaciones que permitiesen comprobar el funcionamiento del programa. Se juzgó conveniente que los grupos fueran divergentes en su función y que no se encuentren relacionados entre sí a nivel de secuencia. Para conformar dichos grupos, las secuencias fueron obtenidas en formato Fasta de la base de datos Pfam [PFAM] . Los grupos se identificaron como Bromodominio, Ciclina, Glutation S-transferasa y TFIIB. Los dominios de estas secuencias representan grupos de proteínas de interés para distintos procesos vitales de los parásitos como son los mecanismos de defensa al estrés oxidante (GST), de regulación del ciclo celular (Ciclina), de regulación basal de la transcripción (TFIIB) y de remodelación de la cromatina (Bromodominio). Las GSTs, por ejemplo, constituyen una familia de proteínas homogénea en cuanto al tamaño, estructura y función, con dos dominios presentes en todos los miembros de la familia. La familia con dominio TFIIB, se encuentra formada por tres grupos de tamaño distinto, que poseen una repetición divergente del dominio. La familia de las Ciclinas es una familia de proteínas con gran variedad de tamaño con un único dominio, el cual además se encuentra lejanamente relacionado con el dominio TFIIB. Finalmente, las proteínas con Bromodominio forman una familia completamente divergente de proteínas con distintas funciones, que sólo se encuentran emparentadas por la presencia del dominio.

Seguidamente se describe brevemente cómo se obtuvieron los grupos de secuencias a analizar y las características de cada uno de ellos.

[7] Lic. Leonardo Ornella (IBR-FCEIA, UNR), comunicación privada

### 4.2.2 Procesamiento de los grupos de secuencias

Inicialmente cada una de las secuencias de los diferentes grupos fueron obtenidas en formato Fasta de Pfam [PFAM] . Pfam es una base de datos de perfiles tipo HMM, que contiene más de siete mil familias de proteínas. Pfam trabaja con dominios, es decir, cada perfil HMM se corresponde con un dominio, aunque no necesariamente cumplan la definición de dominio estructural independiente, sino más bien suelen ser regiones características de una determinada familia de proteínas. Además de las ventajas que de por sí tiene esta clasificación, Pfam resulta útil para: analizar las alineaciones múltiples que contiene, estudiar la organización de dominios de las proteínas, examinar la distribución filogenética de las proteínas que presentan el dominio, y ver la estructura tridimensional de los dominios, cuando ésta se conoce.

A continuación se procesaron los grupos Ciclina y GST utilizando el programa MEGA3 (Kumas, S; Tamura,K; Nei, M [MEGA] ) para obtener una matriz de distancias entre las proteínas asociadas a cada uno de los dominios. A su vez, dichas matrices fueron utilizadas para obtener 2 subgrupos parciales de secuencias, de tal manera que las distancias entre las secuencias de cada subgrupo fueran mayores o menores a un determinado valor límite. Este valor límite depende de la matriz de distancias original y se selecciona de tal manera que se asegure un mínimo de proteínas en cada subgrupo. De esta forma se garantiza que las distancias dentro de cada uno de estos subgrupos son similares. El objetivo de dicha partición fue analizar el comportamiento del programa bajo distintas condiciones de trabajo. Aprovechando la estructura de grafo de la matriz de distancias, los distintos subgrupos fueron obtenidos utilizando una versión de prueba del programa Ucinet para Windows versión 6.59 (Borgatti, S.P., Everett, M.G. y Freeman, L.C. 2002. Ucinet for Windows: Software for Social Network Analysis. Harvard, MA: Analytic Technologies). Los grupos Bromodominio y TFIIB no fueron sometidos a este estudio. Las secuencias del grupo TFIIB fueron elegidas con el objetivo de exhibir en esta tesis los resultados de la comparación de los resultados de diferentes programas, debido a que su motivo es pequeño con relación al de los demás grupos analizados.

### 4.2.3 Detalle de los grupos de secuencias

## BROMODOMINIO

El bromodominio se trata de un motivo de aproximadamente 110 residuos de longitud que se encuentran en una gran variedad de proteínas de mamíferos, invertebrados y levaduras. En un principio estas proteínas fueron aisladas de la cromatina, sin embargo actualmente se conocen proteínas con bromodominio que no se encuentran en el núcleo. Si bien se desconoce la función precisa de los mismos, se han descripto que participan en interacciones proteína-proteína y que juegan un rol importante en el ensamblado y/o actividad de complejos multicomponentes involucrados en la maquinaria transcripcional. Recientemente se ha postulado que la acetilación de proteínas, proceso en el cual las proteínas con bromodominios jugarían un rol esencial, sería un nuevo mecanismo de señalización intracelular. De este modo el estudio de las proteínas con bromodominio puede aportar nuevos datos sobre los mecanismos de transducción de señal que culminan con la expresión diferencial de genes diana. Es de notar que en cada secuencia puede aparecer uno o más bromodominio. Sin embargo, se recuerda que el programa GibbsSM sólo localiza un único motivo por secuencia. En total fueron analizadas 67 secuencias para el grupo bromodominio, de las cuales 17 fueron eliminadas por estar duplicadas. Las restantes 50 secuencias bromodominio tienen longitudes que varían de 300 a 3200 aminoácidos.
La longitud del motivo fue determinada en 110 residuos en la alineación que dispone Pfam.

## GLUTATION S-TRANSFERASA (N-terminal)

Estas proteínas representan una barrera de protección a la acción de compuestos altamente reactivos (tanto exógenos como endógenos) que por su reactividad pueden producir daño en estructuras celulares. En el caso de los parásitos en particular, es conocido que se trata de organismos con algunas deficiencias en los distintos sistemas enzimáticos de protección al estrés, por lo que el estudio de las GSTs puede develar nuevos posibles blancos para el diseño de drogas terapéuticas. En el caso de *T.cruzi*, no han sido descripto actividades de tipo GST, sin embargo han sido encontradas algunas secuencias con dominios de tipo GST altamente divergentes, que no pueden ser detectados mediante programas de alineamiento global o local comúnmente utilizados. En total fueron analizadas 61 secuencias GST N-terminal para este grupo, cuyas longitudes varían desde 190 a 1264 aminoácidos, sin embargo se observa que en su mayoría tienen longitudes que no superan los 250 residuos.

Se analizó la matriz de distancias entre las secuencias de este grupo que emitió el programa MEGA3. Y se conformaron 2 *subgrupos* los cuales se determinaron a partir de una identidad del 35%.

Subgrupo mayor o igual 35%: Se agruparon 12 de las 61 secuencias que tienen una identidad mayor o igual al 35%.

Subgrupo menor a 35%: Se agruparon 21 de las 61 secuencias que tienen una identidad menor al 35%.

La longitud del motivo fue determinada en 105 residuos por la alineación que dispone Pfam.

## CICLINAS

Las ciclinas y sus reguladores, las ciclinas kinasas son las que permiten la progresión del ciclo celular, hacia la división celular, o bien detienen esta progresión, en el caso en que las condiciones externas o fisiológicas de la célula no sean adecuadas para la división. Han sido encontrados homólogos de ciclinas en varios virus, incluyendo el herpesvirus simiri y el herpes virus asociado al sarcoma de Karposi. Estas proteínas virales difieren de sus contrapartes celulares en que han ganado nuevas funciones y perdido otras, lo cual genera una desregulación del ciclo celular a favor del virus. En el caso de los parásitos unicelulares, como *T. cruzi*, las ciclinas son las que regulan la división celular y por lo tanto el desarrollo de los parásitos. Estos mecanismos de regulación son hasta el momento poco conocidos, ya que *T. cruzi* se desarrolla en distintos huéspedes, por lo cual las señales de regulación y aún las ciclinas involucradas pueden diferir a lo largo del ciclo de vida del parásito. En total fueron analizadas 142 secuencias para este grupo, cuyas longitudes varían desde 200 a 1000 aminoácidos, sin embargo en su mayoría sus longitudes no superan los 400 residuos.
Se analizó la matriz de distancias entre las secuencias de este grupo que emitió el programa MEGA3. Y se conformaron 2 *subgrupos* en función del dominio carboxilo terminal, los cuales se determinaron a partir de una identidad del 23%.

Subgrupo mayor o igual 23%: Se agruparon 23 de las 142 secuencias de ciclinas cuyos dominios carboxilo terminal tienen una identidad mayor o igual al 23%.

Subgrupo menor a 23%: Se agruparon 24 de las 142 secuencias de ciclinas cuyos dominios carboxilo terminal tienen una identidad menor al 23% .

La longitud del motivo fue determinada en 122 residuos por el alineamiento disponible en Pfam.

## TFIIB

En eucariotas, la transcripción es llevada a cabo por tres ARN polimerasas diferentes, ARN polimerasa I, II y III, cada una de las cuales es responsable de la transcripción de un grupo diferente de genes. La selección de los genes transcriptos se realiza a través de secuencias promotoras características para cada una de las ARN polimerasas. Se supone que el factor de transcripción TFIIB es de central importancia en la transcripción de genes de clase II. Junto con otros factores TFIIA, TFIID, TFIIE, TFIIF, TFIIG y TFIIH, interaccionan con regiones especificas del ADN regulando la iniciación de la transcripción de la ARN-polimerasa II. El factor de transcripción IIB se asocia con TFIID y TIIFA unidos al ADN para formar un complejo ternario TFIID-IIA-IBB (DAB) el cual es reconocido por la ARN-polimerasa. El dominio TFIIB, también se encuentra en uno de los componentes de TFIIIB, un factor basal específico de la ARN polimerasa III y en TFB un factor de transcripción de Archeabacterias. Esto hace al dominio TFIIB extremadamente interesante desde el punto de vista evolutivo, ya que se trata de uno de los dominios relacionados con la regulación de la transcripción más antigua.

Este dominio, también ha sido relacionado con el dominio ciclina, sobre todo en secuencias de TFIIIB altamente divergentes. Esto tiene sentido biológico ya que TFIIIB podría estar regulado durante el ciclo celular y podría significar que estos dos grupos de proteínas altamente divergentes entre sí podrían haberse originado a partir de un grupo común de proteínas ancestrales. De ser así, los dominios ciclina y TFIIB serían la clave para el establecimiento y el desarrollo del ciclo celular. Para este estudio sólo se propusieron 23 secuencias de la familia de factores de transcripción TFIIB para ser analizadas. La longitud del motivo fue establecida en 10 residuos [8].

[8] Lic. Leonardo Ornella (IBR-FCEIA, UNR), comunicación privada

## 4.3 Resultados Experimentales

A los efectos de evaluar la consistencia de resultados arrojados por GibbsSM, se realizaron pruebas adicionales con dos paquetes de software de uso estándar para el problema de detección de motivos: GIBBS [GIBBS] y MEME [MEME] . La elección de GIBBS fue motivada porque usa la metodología de esta tesis [32][46]. La elección de MEME, en cambio, respondió a que su uso está muy difundido en el ámbito biológico. A diferencia de GibbsSM, GIBBS puede localizar varios motivos en cada una de las secuencias y permite realizar fragmentación del motivo. Sin embargo, para que la comparación sea fiable se solicitó un único motivo por secuencia. A su vez MEME requiere, al igual que GibbsSM, que se prefije el ancho del motivo y no considera ni inserciones ni blancos en el motivo. Para más detalles de los mismos remitirse a la sección 3.6. donde se realizó una descripción de los mismos.

Cabe aclarar que los detalles de los resultados de los grupos Bromodominio, Glutation S-transferasa y Ciclina no se incluyen en esta presentación por ser los motivos de estos grupos muy extensos: 110, 105 y 122 respectivamente. No obstante, se encuentran disponibles en la página web de esta tesis http://www.eie.fceia.unr.edu.ar/~bioinfo/gibbs y se analizan en la siguiente sección.

Los resultados de los motivos del grupo de secuencias de factores de transcripción TFIIB son los que se eligieron para ilustrar esta tesis.

## TFIIB

Las condiciones iniciales para el grupo de secuencias del factor de transcripción TFIIB, incluyen las 23 secuencias en formato FASTA, un ancho de motivo en 10, el tipo de secuencia: proteínas, el valor de la semilla de aleatorización: 12345, la proporcionalidad de pseudocount: 0.001, la cantidad de iteraciones = 4 (para detener el proceso de muestreo Gibbs), la cantidad máxima de lazos de repetición para calcular el MAP= 5, el error máximo entre dos aproximaciones del MAP= 0.001, y no se reordenaron las secuencias (shuffle en false)

**Condiciones iniciales**

```
File of sequences = T23.txt
# of sequences = 23
motif width    = 10
# alpha        = 21 (alphabet <ARNDCQEGHILKMFPSTWYVX>)
random seed    = 12345
Pseudocount ratio = 0.0010
Runs =  4
Iteration 5 loops
Epsilon = 1.0E-5
Shuffle false
```

## **Resultado del** *GibbsSM*

En la Tabla 4.1 se observa la alineación de los motivos de las secuencias del grupo TFIIB que se obtuvo con el programa GibbsSM. Este resultado es mostrado en pantalla y a su vez grabado en un archivo con igual nombre al de las secuencias pero con extensión *gibbs*. Este archivo está disponible en la página web http://www.eie.fceia.unr.edu.ar/~bioinfo/gibbs. El programa GibbsSM muestra el alineamiento de las secuencias y a continuación el modelo estadístico de este perfil constituido por la matriz de probabilidades del motivo y el vector de probabilidades del background, valores que se muestran en la Tabla 4.2. Finalmente dispone del log-likelihood ratio del mejor alineamiento encontrado y el tiempo de ejecución.

Tabla 4.1. Alineamiento de los motivos del grupo *TFIIB* dado por GibbsSM

| **Sequences** | | **pos_ini** | **sites** | **pos_end** | |
|---|---|---|---|---|---|
| Homo2B | lkgrandaia | 161 | SACLYIACRQ | 170 | egvprtfkei |
| Homo3B | trgrkmahvi | 133 | AACLYLVCRT | 142 | egtphmlldl |
| Dros2B | lkgrsndaka | 160 | SACLYIACRQ | 169 | egvprtfkei |
| Dros3B | trgrksthiy | 134 | AACVYMTCRT | 143 | egtshllidi |
| Arab2Ba | srgrnqdall | 152 | AACLYIACRQ | 161 | edkprtvkei |
| Arab2Bb | rrgkklnaic | 162 | AASVSTACRE | 171 | lqlsrtlkei |
| Soy2B | srgrnqdall | 153 | AACLYIACRQ | 162 | edkprtvkei |
| Caenoel2B | lrgknneaqa | 151 | AACLYIACRK | 160 | dgvprtfkei |
| Xenolae2B | lkgrsndaia | 161 | SACLYIACRQ | 170 | egvprtfkei |
| Candal3B | vqgrrsnnvl | 135 | ATCLYVACRK | 144 | erthhmlidf |
| Schizpom2B | lkgkssqsii | 165 | AACIYIACRQ | 174 | gkvprtfmei |
| Saccer2B | lkgksmesim | 173 | AASILIGCRR | 182 | aevartfkei |
| Saccer3B | vqgrrsqnvi | 127 | ASCLYVACRK | 136 | ekthhmlidf |
| Ghill2B1 | vinkkvdlyi | 117 | ISCLYMISRF | 126 | ektphllvdf |
| Ghill2B2 | mygrnyisva | 256 | AASIYVVSQI | 265 | pnlsnncnlk |
| Dyctio2B | tkgrqtrlva | 121 | AACLYIVCRR | 130 | ertphllidf |
| Metja2B | irgrsiegvv | 532 | AAAIYAACRR | 541 | crvprtldei |
| Metther2B | irgrsiegvv | 169 | AASLYAACRK | 178 | cnvprtldei |
| Aeroper2B | trgrsiesiv | 170 | AASLYAASRI | 179 | hglphsltdi |
| Sulfshi2B | vrgrsiesvv | 167 | AAAIYAACRR | 176 | mklartldei |
| Archful2B | irgrsiegvv | 185 | AAALYAACRQ | 194 | agvprtldei |
| Pyroc2B | vrgrslesma | 168 | AAAVYAACRI | 177 | rgiprsiddi |
| Giardia | trgrrnnlla | 116 | AALLYIVGRQ | 125 | hnlshllidy |

En la primera columna de la Tabla 4.1 se observan los nombres abreviados de las secuencias analizadas. La tercera y quinta columnas se indica donde comienza y termina el motivo dentro de la secuencia (en número de residuos). En la segunda y sexta columna se exhiben los 10 residuos

anteriores y posteriores al motivo. Mientras que, en la cuarta columna, y en mayúsculas se muestra el motivo en cada una de las 23 secuencias.

Tabla 4.2: Perfil del motivo del grupo *TFIIB* con GibbsSM

Matriz de probabilidades del motivo

| #pos | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0,608 | 0,639 | 0,147 | 0,024 | 0,024 | 0,208 | 0,516 | 0,024 | 0,024 | 0,024 |
| C | 0,006 | 0,006 | 0,406 | 0,006 | 0,006 | 0,006 | 0,006 | 0,590 | 0,006 | 0,006 |
| D | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 |
| E | 0,024 | 0,024 | 0,024 | 0,024 | 0,024 | 0,024 | 0,024 | 0,024 | 0,024 | 0,055 |
| F | 0,008 | 0,008 | 0,008 | 0,008 | 0,008 | 0,008 | 0,008 | 0,008 | 0,008 | 0,039 |
| G | 0,018 | 0,018 | 0,018 | 0,018 | 0,018 | 0,018 | 0,048 | 0,048 | 0,018 | 0,018 |
| H | 0,005 | 0,005 | 0,005 | 0,005 | 0,005 | 0,005 | 0,005 | 0,005 | 0,005 | 0,005 |
| I | 0,052 | 0,021 | 0,021 | 0,175 | 0,021 | 0,328 | 0,052 | 0,021 | 0,021 | 0,113 |
| K | 0,023 | 0,023 | 0,023 | 0,023 | 0,023 | 0,023 | 0,023 | 0,023 | 0,023 | 0,146 |
| L | 0,025 | 0,025 | 0,056 | 0,486 | 0,056 | 0,056 | 0,025 | 0,025 | 0,025 | 0,025 |
| M | 0,006 | 0,006 | 0,006 | 0,006 | 0,006 | 0,068 | 0,006 | 0,006 | 0,006 | 0,006 |
| N | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 |
| P | 0,011 | 0,011 | 0,011 | 0,011 | 0,011 | 0,011 | 0,011 | 0,011 | 0,011 | 0,011 |
| Q | 0,009 | 0,009 | 0,009 | 0,009 | 0,009 | 0,009 | 0,009 | 0,009 | 0,040 | 0,255 |
| R | 0,020 | 0,020 | 0,020 | 0,020 | 0,020 | 0,020 | 0,020 | 0,020 | 0,696 | 0,143 |
| S | 0,115 | 0,084 | 0,176 | 0,023 | 0,054 | 0,023 | 0,023 | 0,115 | 0,023 | 0,023 |
| T | 0,015 | 0,046 | 0,015 | 0,015 | 0,015 | 0,046 | 0,046 | 0,015 | 0,015 | 0,077 |
| V | 0,017 | 0,017 | 0,017 | 0,109 | 0,017 | 0,109 | 0,140 | 0,017 | 0,017 | 0,017 |
| W | 0,001 | 0,001 | 0,001 | 0,001 | 0,001 | 0,001 | 0,001 | 0,001 | 0,001 | 0,001 |
| Y | 0,007 | 0,007 | 0,007 | 0,007 | 0,653 | 0,007 | 0,007 | 0,007 | 0,007 | 0,007 |
| X | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |

Probabilidades del Background

```
A: 0,0769  C: 0,0174  D: 0,0604  E: 0,0834  F: 0,0286  G: 0,0602
H: 0,0155  I: 0,0719  K: 0,0803  L: 0,0860  M: 0,0207  N: 0,0467
P: 0,0373  Q: 0,0304  R: 0,0666  S: 0,0786  T: 0,0523  V: 0,0582
W: 0,0054  Y: 0,0230  X: 0,0000
```

```
log-likelihood ratio = 534.1055127615554
Tiempo de ejecución =  2 segundos
```

## Resultado del GIBBS

En la Tabla 4.3 se observa la alineación de los motivos de las secuencias del grupo TFIIB obtenidas con el programa de Rouchka *et al*. disponible en la web [GIBBS] al momento del desarrollo de esta tesis. El resultado de este programa es enviado por e-mail al destinatario que solicitó su servicio, el mensaje original está disponible en la página web http://www.eie.fceia.unr.edu.ar/~bioinfo/gibbs. Este mensaje contiene las condiciones iniciales, el alineamiento de los motivos y el modelo estadístico constituido por la matriz de probabilidades del motivo y el vector de probabilidades del background, valores que se muestran en la Tabla 4.4. Finalmente dispone del valor del MAP y el tiempo de ejecución.

Tabla 4.3: Alineamiento de los motivos del grupo *TFIIB* dado por GIBBS

| # | pos_ini | Motif Element | pos_end | Sequences |
|---|---|---|---|---|
| 1, | 161 | ndaia SACLYIACRQ egvpr | 170 | Homo2B |
| 2, | 133 | mahvi AACLYLVCRT egtph | 142 | Homo3B |
| 3, | 160 | ndaka SACLYIACRQ egvpr | 169 | Dros2B |
| 4, | 134 | sthiy AACVYMTCRT egtsh | 143 | Dros3B |
| 5, | 152 | qdall AACLYIACRQ edkpr | 161 | Arab2Ba |
| 6, | 162 | lnaic AASVSTACRE lqlsr | 171 | Arab2Bb |
| 7, | 153 | qdall AACLYIACRQ edkpr | 162 | Soy2B |
| 8, | 151 | neaqa AACLYIACRK dgvpr | 160 | Caenoel2B |
| 9, | 161 | ndaia SACLYIACRQ egvpr | 170 | Xenolae2B |
| 10, | 135 | snnvl ATCLYVACRK erthh | 144 | Candal3B |
| 11, | 165 | sqsii AACIYIACRQ gkvpr | 174 | Schizpom2B |
| 12, | 173 | mesim AASILIGCRR aevar | 182 | Saccer2B |
| 13, | 127 | sqnvi ASCLYVACRK ekthh | 136 | Saccer3B |
| 14, | 117 | vdlyi ISCLYMISRF ektph | 126 | Ghill2B1 |
| 15, | 256 | yisva AASIYVVSQI pnlsn | 265 | Ghill2B2 |
| 16, | 121 | trlva AACLYIVCRR ertph | 130 | Dyctio2B |
| 17, | 532 | iegvv AAAIYAACRR crvpr | 541 | Metja2B |
| 18, | 169 | iegvv AASLYAACRK cnvpr | 178 | Metther2B |
| 19, | 170 | iesiv AASLYAASRI hglph | 179 | Aeroper2B |
| 20, | 167 | iesvv AAAIYAACRR mklar | 176 | Sulfshi2B |
| 21, | 185 | iegvv AAALYAACRQ agvpr | 194 | Archful2B |
| 22, | 168 | lesma AAAVYAACRI rgipr | 177 | Pyroc2B |
| 23, | 116 | nnlla AALLYIVGRQ hnlsh | 125 | Giardia |

La primer columna indica el número y la última el nombre de cada una de las secuencias del grupo TFIIB. La segunda y la sexta columna señalan donde comienza y finaliza el motivo dentro de la secuencia (en número de residuos). La tercera y quinta columnas indican los 5 residuos anteriores y posteriores al motivo. La cuarta, en letras mayúsculas, está resaltado el motivo en cada secuencia.

Luego del alineamiento, se muestra la matriz del modelo de probabilidades del motivo, tanto de los motivos como del background.

Tabla 4.4: Perfil del motivo del grupo *TFIIB* con GIBBS

Matriz de probabilidades del motivo

| Pos. # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.758 | 0.798 | 0.166 | 0.007 | 0.007 | 0.245 | 0.640 | 0.007 | 0.007 | 0.007 |
| C | 0.002 | 0.002 | 0.516 | 0.002 | 0.002 | 0.002 | 0.002 | 0.753 | 0.002 | 0.002 |
| D | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 |
| E | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.047 |
| F | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.042 |
| G | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.005 | 0.045 | 0.045 | 0.005 | 0.005 |
| H | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| I | 0.046 | 0.007 | 0.007 | 0.204 | 0.007 | 0.402 | 0.046 | 0.007 | 0.007 | 0.125 |
| K | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.165 |
| L | 0.008 | 0.008 | 0.047 | 0.601 | 0.047 | 0.047 | 0.008 | 0.008 | 0.008 | 0.008 |
| M | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 | 0.081 | 0.002 | 0.002 | 0.002 | 0.002 |
| N | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 |
| P | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 |
| Q | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.042 | 0.319 |
| R | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.876 | 0.164 |
| S | 0.126 | 0.086 | 0.205 | 0.007 | 0.047 | 0.007 | 0.007 | 0.126 | 0.007 | 0.007 |
| T | 0.005 | 0.044 | 0.005 | 0.005 | 0.005 | 0.044 | 0.044 | 0.005 | 0.005 | 0.084 |
| V | 0.005 | 0.005 | 0.005 | 0.124 | 0.005 | 0.124 | 0.163 | 0.005 | 0.005 | 0.005 |
| W | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Y | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 | 0.002 |

Probabilidades del Background

```
A: 0,0770  C: 0,0174  D: 0,0604  E: 0,0834  F: 0,0286  G: 0,0602
H: 0,0155  I: 0,0719  K: 0,0803  L: 0,0860  M: 0,0207  N: 0,0467
P: 0,0373  Q: 0,0304  R: 0,0666  S: 0,0786  T: 0,0523  V: 0,0582
W: 0,0054  Y: 0,0230  X: 0,0000
```

```
Log Motif portion of MAP for motif a = -310.31385 (log negativo )
Elapsed time: 6.980000 secs
```

## Resultado del MEME

En la Tabla 4.5 se observa la alineación de los motivos de las secuencias del grupo TFIIB obtenidas con el MEME disponible al momento del desarrollo de esta tesis [MEME] . El resultado de este programa es enviado por e-mail al destinatario que solicitó su servicio, el mensaje original está disponible en la página web http://www.eie.fceia.unr.edu.ar/~bioinfo/gibbs. Este mensaje contiene las secuencias que se analizan, las condiciones iniciales, el alineamiento de los motivos y, además las secuencias aparecen listadas en orden creciente de la significancia estadística del motivo (*p-value*) [6][30]. El *p-value* da la probabilidad de encontrar un alineamiento con un score igual o mejor a un score dado. El valor de *p-value* es calculado relacionando el score del alineamiento observado con la distribución de scores esperada calculada sobre un conjunto de secuencias aleatorias de igual longitud y composición como la secuencia que se está analizando. Para mayor detalles del *p-value* remitirse a Bailey *et al.* 1998[6] y Karlin *et al.* 1990 [30]. Finalmente se muestran, en forma gráfica, los motivos encontrados en cada una de las secuencias.

Tabla 4.5. Alineamiento de los motivos del grupo *TFIIB* dado por MEME

| NAME | START | P-VALUE | SITES |
|---|---|---|---|
| Soy2B | 153 | 7.15e-14 | SRGRNQDALL AACLYIACRQ EDKPRTVKEI |
| Arab2Ba | 152 | 7.15e-14 | SRGRNQDALL AACLYIACRQ EDKPRTVKEI |
| Schizpom2B | 165 | 2.33e-13 | LKGKSSQSII AACIYIACRQ GKVPRTFMEI |
| Caenoel2B | 151 | 6.98e-13 | LRGKNNEAQA AACLYIACRK DGVPRTFKEI |
| Xenolae2B | 161 | 1.29e-12 | LKGRSNDAIA SACLYIACRQ EGVPRTFKEI |
| Dros2B | 160 | 1.29e-12 | LKGRSNDAKA SACLYIACRQ EGVPRTFKEI |
| Homo2B | 161 | 1.29e-12 | LKGRANDAIA SACLYIACRQ EGVPRTFKEI |
| Archful2B | 185 | 8.01e-12 | IRGRSIEGVV AAALYAACRQ AGVPRTLDEI |
| Dyctio2B | 121 | 8.49e-12 | TKGRQTRLVA AACLYIVCRR ERTPHLLIDF |
| Metther2B | 169 | 4.21e-11 | IRGRSIEGVV AASLYAACRK CNVPRTLDEI |
| Saccer3B | 127 | 7.43e-11 | VQGRRSQNVI ASCLYVACRK EKTHHMLIDF |
| Candal3B | 135 | 9.56e-11 | VQGRRSNNVL ATCLYVACRK ERTHHMLIDF |
| Sulfshi2B | 167 | 1.29e-10 | VRGRSIESVV AAAIYAACRR MKLARTLDEI |
| Metja2B | 532 | 1.29e-10 | IRGRSIEGVV AAAIYAACRR CRVPRTLDEI |
| Homo3B | 133 | 1.91e-10 | TRGRKMAHVI AACLYLVCRT EGTPHMLLDL |
| Dros3B | 134 | 2.26e-10 | TRGRKSTHIY AACVYMTCRT EGTSHLLIDI |
| Pyroc2B | 168 | 4.01e-10 | VRGRSLESMA AAAVYAACRI RGIPRSIDDI |
| Aeroper2B | 170 | 8.89e-09 | TRGRSIESIV AASLYAASRI HGLPHSLTDI |
| Giardia | 116 | 1.91e-08 | TRGRRNNLLA AALLYIVGRQ HNLSHLLIDY |
| Saccer2B | 173 | 3.20e-08 | LKGKSMESIM AASILIGCRR AEVARTFKEI |
| Arab2Bb | 162 | 6.16e-08 | RRGKKLNAIC AASVSTACRE LQLSRTLKEI |
| Ghill2B2 | 160 | 2.00e-07 | LRKKDTFSII AASIFIICKN ESIPRSFKEI |
| Ghill2B1 | 117 | 5.09e-07 | VINKKVDLYI ISCLYMISRF EKTPHLLVDF |

## 4.4 Análisis de los resultados

Este análisis se centra en la comparación de los resultados que se obtuvieron con el programa GibbsSM y los programas MEME y GIBBS. En la Tabla 4.6 se muestran en forma simplificada los resultados obtenidos.

Es de aclarar que se referirá como "se observa un *adelantamiento* del sitio del motivo dado por GibbsSM respecto al resultado de ..." significando que en el GibbsSM ese motivo fue localizado en posiciones anteriores a la de comienzo en el otro programa. Recíprocamente, se referirá como "se observa un *atraso* del sitio del motivo dado por GibbsSM respecto al resultado de ..." significando que en el GibbsSM ese motivo fue localizado en posiciones posteriores a la de comienzo en el otro programa. A continuación se refieren comentarios a los resultados observados:

1- **TFIIB:** Los motivos del grupo TFIIB obtenidos con el programa GibbsSM coincidieron exactamente con los dados por el programa de referencia GIBBS. Los valores de los modelos estadísticos de ambas alineaciones, referenciados en Tablas 4.2 y 4.4, muestran una ligera diferencia en los valores numéricos pero no en los pesos de cada aminoácido dentro del perfil. Esta diferencia se puede deber a la inicialización de los pseudocounts en el modelo. En la temporización se obtuvo un mejor tiempo de ejecución con el GibbsSM. Por alguna razón que escapa a este estudio, no se detalla en la documentación del programa GIBBS los valores de pseudocounts utilizados, ni las prestaciones computacionales donde se ejecuta el software. Respecto a los resultados emitidos por el MEME se observa una coincidencia exacta en todas las secuencias salvo en la Ghill2B2 que tiene un corrimiento considerable.

2- **Bromodominio:** Los resultados obtenidos con el GibbsSM en los grupos Bromodominio resultaron sitios adelantados en 15 residuos respecto a los emitidos por MEME, y en 16 residuos respecto de los resultados de GIBBS en las 50 secuencias. Como los motivos buscados son de longitud 110, este avance no es significativo, debido a que el motivo calculado con GibbsSM está incluyendo al calculado por MEME y por GIBBS. Sin embargo, hay que tener en cuenta que estas secuencias pueden tener uno o dos dominios y que tal vez este corrimiento revela la limitación del programa GibbsSM que sólo busca un dominio por secuencia.

3- **Ciclina:** Los resultados obtenidos en el grupo **completo** de Ciclina C-terminal (142 secuencias) resultaron en un corrimiento del dominio hacia adelante de 5 residuos con respecto al establecido por MEME, y en 11 residuos con respecto al dado por GIBBS. En este caso se debe considerar que la longitud del motivo dado por el programa Pfam fue de 122. Es de notar que, en el alineamiento dado por Pfam se observa una gran cantidad de huecos. No obstante, el procesamiento de los subgrupos arrojó resultados más precisos. En el subgrupo **mayor a 23%** (24 secuencias) todos los motivos obtenidos con el GibbsSM se vieron atrasados en 1 residuos respecto a los dados por GIBBS, y atrasados en 1 residuo respecto a los emitidos por MEME. Mientras que en el otro subgrupo, **menor a 23%** (24 secuencias), los motivos obtenidos con el GibbsSM se vieron adelantados en 1 residuo respecto a los dados por GIBBS, y coincidieron exactamente los sitios en 20 de las 24 secuencias respecto al MEME. Se puede deducir que los sitios calculados por GibbsSM son consistentes con los calculados por MEME y tienen mínima variación con respecto a los obtenidos con GIBBS.

4- **GST:** Los resultados obtenidos con el GibbsSM en los grupos **completo** de Glutation S-transferasa (61 secuencias) respecto a los emitidos por MEME, resultaron en un 85% sitios adelantados en 3 residuos y un 15% valores dispares. Respecto a los entregados por GIBBS, los resultados obtenidos en el GibbsSM resultaron sitios atrasados en 3 residuos en el 95% de las secuencias, mientras que el 5% restantes se vieron valores muy dispares. En el subgrupo **mayor a 35%** (12 secuencias) los motivos obtenidos con el GibbsSM se vieron en un 85% adelantados entre 1 a 18 residuos respecto a los dados por GIBBS, el 15% restante atrasados en 17 residuos. Respecto a los emitidos por MEME, los sitios en GibbsSM coincidieron en 5 de las 12 secuencias, 5 secuencias atrasadas en 7 residuos, y en 36 residuos las 2 restantes. En el subgrupo **menor a 35%** (21 secuencias), los motivos obtenidos con el GibbsSM coincidieron en 20 de las 21 secuencias, la restante dio muy dispar. Respecto a los resultados con MEME los sitios se vieron atrasados en 18 residuos respecto de los resultados de GibbsSM en un 60% de las secuencias, mientras que el resto dieron muy dispares. Esta situación se debe a que las secuencias en este grupo son muy divergentes.

5- **Valores de pseudocounts:** Se realizaron pruebas modificando los valores de los pseudocounts en el GibbsSM, con el objetivo de ver como funcionaba el programa cuando se le asigna más peso a un residuo que a otros. El resultado de estas pruebas dieron como resultado motivos donde tuvieron mayor preponderancia los residuos con mayor peso en el modelo.

Basándonos en estas comparaciones, se puede deducir que el programa GibbsSM provee resultados coherentes y racionales dentro del marco teórico desarrollado.

Tabla 4.6 Resultados comparativos GibbsSM vs.GIBBS y GibbsSM vs. MEME

| **Grupo** | **Distancia** | **Ancho del motivo** | **Offset † respecto a GIBBS** | **Offset † respecto a MEME** |
|---|---|---|---|---|
| *TFIIB* | NA | 10 | 0 | 0 |
| *Bromodominio* | NA | 110 | -16 | -15 |
| *Ciclina* | NA | 122 | -11 | -5 |
| | mayor 23% | 122 | +1 | +1 |
| | menor 23% | 122 | +1 | 0 |
| *GST* | NA | 105 | +3 ( 95% )<br>NA ( 5% ) | -3 ( 85% ) ‡<br>NA ( 15% ) |
| | mayor 35% | 105 | -18 a -1 ( 85% )<br>+17 ( 15% ) | 0 a +7 ( 85% )<br>NA ( 15% ) |
| | menor 35% | 105 | 0 | -18 ( 60% ) ‡<br>NA ( 40% ) |

† Los valores de offset negativos significan que el sitio del motivo dado en el GibbsSM se inicia en posiciones anteriores a las que de comienzo en los otros programas. Un valor de offset positivo significa que el sitio en GibbsSM comienza en posiciones posteriores a los sitios de comienzo de los otros programas.

‡ Entre paréntesis se indica el porcentaje de secuencias con el offset indicado. Se usa NA para los casos en los cuales no existe una regularidad en los offsets.

# Capitulo 5: Conclusiones

*"La confusión está clarísima"*
*Anónimo*

En esta tesis se analizó la aplicación del muestreo Gibbs para problemas de detección de motivos dentro de secuencias biológicas. El método sólo requiere el conjunto de secuencias y una estimación de la longitud del motivo. A diferencia del EM tiene la ventaja de localizar los motivos en tiempo lineal respecto a la cantidad de secuencias. Usando el muestreo estocástico, el muestreo Gibbs, permite escapar de máximos locales en los cuales las aproximaciones determinísticas suele quedar atrapadas.

A los fines de mostrar las bondades del método en el problema considerado se desarrolló un programa específico denominado *GibbsSM.* Tanto el código como la aplicación están disponibles en la página www.eie.fceia.unr.edu.ar/~bioinfo/gibbs. El programa GibbsSM sólo requiere como información el ancho del motivo y las secuencias que deben estar libres de espacios. Este programa tiene una interfaz de usuario simple para adquirir la información y mostrar los resultados. El funcionamiento del programa se mostró con datos reales obtenidos en el laboratorio del IBR (Serra *et al*. 2003). Estos datos corresponden al parásito unicelular *Trypanosoma cruzi*, responsable de la enfermedad de Chagas. Se analizaron cuatro grupos divergentes entre sí, el Bromodominio, el Glutation S-transferasa, el Ciclina y el TFIIB. Los resultados, que fueron analizados desde un punto de vista computacional como biológico[9], fueron satisfactorios.

La primera conclusión es que el muestreo Gibbs es una alternativa computacional a la metodología EM que permite el manejo coherente de datos experimentales complejos. Además, el método se caracteriza por su gran simplicidad y escalabilidad respecto al número de variables (ver sección 3.4).

[9] Lic. Leonardo Ornella (IBR-FCEIA, UNR) private comunication

La segunda conclusión es que el muestreo Gibbs es de formulación simple pero su aplicación exitosa depende de un correcto modelado estadístico del problema bajo estudio. Obviamente la percepción de correcto tiene que ver con el balance, complejidad y exactitud del modelo estadístico propuesto. La metodología permite el muestreo de *cualquier* distribución de probabilidad a posteriori, incluso aquellas con datos ocultos. La complejidad de tales modelos y correspondientes parámetros se suaviza mediante la introducción de conocimiento a priori bajo la forma de distribuciones de probabilidad en contexto completamente bayesiano. Más aún, la elección de conocimiento previo bajo distribución de probabilidades a priori es primordial. Esta alternativa es fundamental en situaciones de inferencia complejas como aquellas asociadas a problemas biológicos.

Con relación al trabajo futuro y en el contexto de continuidad del trabajo conjunto con el IBR, se prevé el modelado de las dependencias dentro de los motivos. Este tipo de problemas es particularmente interesante con relación a los factores de transcripción y puede dar lugar a hipótesis biológicas importantes. De forma similar, mencionamos que la aplicación del muestreo Gibbs en Bioinformática ha trascendido el problema de detección de motivos, para aplicarse a la nueva generación de problemas referidos a la reconstrucción parcial de haplotipos [10], como se observa en el trabajo de Niu *et al.* 2002. En cualquier caso, es mi deseo que este trabajo de tesis permita un abordaje rápido de estos y otros problemas en Bioinformática que puedan ser solucionables con muestreo Gibbs.

[10] Los haplotipos son marcadores genéticos en un cromosoma

# Referencias bibliográficas

[1] Altshul, S., Gish, W., Miller, W., Myers W. and Lipman, D., *Basic local alignment search tool,* Journal of Molecular Biology, 215: 403-410, 1990.

[2] Altschul, S., Madden, T. Schaeffer, T. Zhang,J. Zhang,Z. Miller,W. and Lipman,D. *Gapped BLAST and PSI-BLAST: A new generation of protein database search programs*. Nucleic Acids Res., 25: 3899-3402, 1997.

[3] Baldi, P. – S. Brunak, *Bioinformatics, The machine learning approach* – Second Edition – MIT- 2001.

[4] Bailey, David, Tesis PhD. *Bayesian Methods for Adaptive Models*, 1992.

[5] Bailey, Timothy and Elkan, Charles. "Fitting a mixture model by expectation maximization to discover motifs in biopolymers", Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28-36, AAAI Press, Menlo Park, California, 1994.

[6] Bailey, T. y Gribskov, M. *Combining evidence using p-values: application to sequence homology searches*, Bioinformatics, 14(48-54), 1998.

[7] Bayes, Thomas, *An Essay towards solving a Problem in the Doctrine of Chances*, Philosophical Transactions of the Royal Society of London, 330-418, 1763.

[8] Bernardo, J. M. y Smith, A. F. M., *Bayesian Theory.* John Wiley and Sons, Ltd., Chichester, 1994.

[9] Bucher, P. *A generalised profile syntax for protein and nucleic acid sequence motifs* Version 1.31, Swiss Institute of Bioinformatics, 2001.

[10] Carrillo, H. y Lipmann, D. *The multiple sequence alignment problem in biology.* SIAM J. Appl.Math, 48:1073-1082, 1988.

[11] Cowles, M.K. y Carlin, B.P., *Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review*. Journal of the American Statistical Association, 91: 883-904, 1996.

[12] Dayhoff, M -R. Schwartz y B. Orcutt, *A model of evolutionary change in proteins.* National Bio. Res. Foundation, Washington DC, 345-352, 1978.

[13] Demspter, A. P., Laird, N. M., Rubin, D. B. *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society B, 39(1): 1:38, 1977.

[14] Dopazo, J. y Valencia, A., *Bioinformática y genómica*, 2001.

[15] Durbin - Eddy - Krogh – Mitchison. *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*. Univ. Cambridge, United Kingdom, 2002.

[16] Gelfand, A. E. y Smith, A. F. M., *Sampling-based approaches to calculating marginal densities*. Journal of the American Statistical Assoc, 85, 398-409. 1990.

[17] Gelman, A., Roberts, G., y Gilks, W. *Efficient Metropolis jumping rules. In Bayesian Statistics 5.* Oxford University Press, New York, 1995.

[18] Gelman, A. y Rubin, D.B., *Inference from Iterative Simulation Using Multiple Sequences*, (with discussion). Statistical Science, 7,457-511, 1992.

[19] Geman, S. y Geman, D. *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intel 6, 721-41, 1984.

[20] Geyer, C.J., *Practical Markov Chain Monte Carlo.* Statistical Science, 7, 473-483, 1992.

[21] Gribskov, M., Luthy,R. y Eisenberg, D. , *Profile analysis,* Methods in Enzymology 183: 146-159, 1990.

[22] Gribskov, M, McLachlan, A, y Eisenberg, D.. *Profile analysis: detection of distantly related proteins,*National Academy of Sciences of the USA 84: 4355-4358, 1987.

[23] Hastings, W.K. *Monte Carlo Sampling Methods Using Markov Chains and ThierApplications*. Biometrika, Vol.57,1970.

[24] Hildebrand, D y Lyman, R. *Estadística aplicada,* Prentice Hall, 1998.

[25] Henikoff, S. y J.G. Henikoff, *Amino acid substitution matrices from protein blocks.* National Academy of Sciences of the USA, 89:10915-10919, 1992.

[26] Higgins, D. Bleasly, A. Fuchs, R. *CLUSTAL V: Improved Software for Multiple Sequence Alignment,* CABIOS 8, 189-191, 1991.

[27] Ide, J.S., *Raciocnio Probabilstico em Sistemas Embarcados*. Anais do XXI Congresso da Sociedade Brasileira de Computao, Ed. Ana Teresa Martins, Fortaleza, 2001.

[28] Karchin, R. *Hidden Markov Models and Protein Sequence Analysis*, Compbio at Oak Ridge National Laboratory, 1999.

[29] Karlin, S. y Altschul, S., *Applications and ststictics for multiple high-scoring segments in molecular sequences*. Prc. Natl. Acad. Sci. USA, Vol 90,pp.5873-5877, 1993.

[30] Karlin, S. y Altschul, S.F *Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes.* Proc. Natl. Acad. Sci. USA 87:2264-2268, 1990.

[31] Krogh, A., M. Brown, I. S. Mian, K. Sjolander, y D. Haussler, *Hidden Markov models in computational biology: Applications to protein modeling*. Journal of Molecular Biology , 235:1501-1531, 1994.

[32] Lawrence, C.E., Altschul, S., M. S. Boguski, J.S. Lui, A.F. Neuwald y J.C. Woottcen, J.C., *Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment*. Science 262:201-208, 1993.

[33] Li, K.H., *Imputation Using Markov Chains*, Journal of Statistical Computation and Simulation, 30, 57 –79, 1988.

[34] Lipman, D.J. y Pearson, W. Rapid and sensitive protein similarity searches. Science,227:1435–1441, 1985.

[35] Lipman, D.J. y Pearson, W. Improved tools for biological *sequence comparison*. Proceedings of the National Academy of Science USA, 85:2444–2448, 1988.

[36] Lipman, D., Altschul, S. y Kececioglu, J., *A tool for multiple sequence aligment*, Nat. Acad. Of Sciences of USA, 86: 4412-4415, 1989.

[37] Liu, Jun, *Monte Carlo Strategies in Scientific Computing*, Springer, 2001.

[38] Liu, Jun , *The collapsed Gibbs sampler with applications to a gene regulation problem*. J.Am.Statist:Assoc. 1994.

[39] Liu, Jun , Neuwald, A.F. y Lawrence, C.E., *Bayesian models for multiple local sequence alignment and Gibbs* sampling strategies. J. Amer. Statistical Assoc. 90:1156-1169, 1995.

[40] Liu, J., Neuwald, A., Lawrence, C.E., *Markovian Structures in Biological Sequence Alignments*, J. Amer. Stat. Assoc., 94, 1-15, 1999.

[41] Lodish, Berk, Zipursky, Matsudaira, Baltimore y Darnell. *Biología Celular y Molecular. (4ta. edición)* Ed. Médica Panamericana, 2002.

[42] MacKay, D., *Information Theory, Inference, and Learning Algorithms*, Univ. Cambridge, United Kingdom, 2003.

[43] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. y Teller, E. *Equation of State Calculations by Fast Computing Machines*, J. Chem. Phys.1953.

[44] M. Li, B. Ma, and L. Wang, *Finding Similar Regions in Many Sequences*, J. Comput. Syst. Sci. 65(1): 73-96 (2002)

[45] M. Li, B. Ma, and L. Wang, *On The Closest String and Substring Problems,* J. Comput. Science CE/0002012 (2000)

[46] Neuwald, A.F., Liu, J.S. y Lawrence, *Gibbs motif sampling: detection of bacterial outer membrane protein repeats* Protein Science 4, 1618-1632, 1995.

[47] Needleman, S. B. y Wunsch, C. D. *A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins*. Journal of Molecular Biology, 48:443-453, 1970.

[48] Niu, T., Qin, Z., Xu,X. y Liu, J. *Bayesian Haplotype Inference for Multiple Linked Single-Nucleotide Polymorphisms*, Am. J. Hum. Genet. 70:157–169, 2002.

[49] Orchard y M. A. Woodbury. *A missing information principle: Theory and applications*. In L. M. Le Cam, J. Neyman, y E. L. Scott, editors, Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 697-715. University of California Press, 1972.

[50] Pohl, F., Nature New Biol. 234,277 (1971); O. Berg y P. Von Hippel, J. Mol. Biol. 193, 723 (1987); S. Bryant and C.Lawrence, Proteins 16, 92, 1993.

[51] Rabiner, L. R. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2):257–286, 1989.

[52] Ramos, F, Cozman, F, Ide, J, *Embedded bayesian Networks: Anyspace,Anytime Probabilistic Inference*. Joint Workshop on Real-Time Decision Support and Diagnosis Systems, AAAI Press, California, 2002.

[53] Rocke, Emile y Martin Tompa, *An Algorithm for Finding Novel Gapped Motifs in DNA Sequences*, 1999.

[54] Richardson, J. Adv. Protein Chem. 34,167 (1981); C. Chorhia, Annu.Rev.Biochem.53, 537 (1984); A. Sali and T. Blundell, J.Mol.Biol. 212,403 (1990); R. Doolittle, Protein Sci. 1, 191, 1992.

[55] Shamir, Ron, Curso *Algorithms in Molecular Biology*, Lecture 2 y 3 en http://www.math.tau.ac.il/~rshamir, Tel Aviv University School of Computer Science, 2002.

[56] Serra, E., Tapia, E., Blet, N. y Nocito,I., "Caracterización de factores basales de transcripcion en parásitos protozoarios" Instituto de Biología Molecular y Celular de Rosario, CONICET - Facultad de Ciencias Bioquímicas y Farmacéuticas, UNR Resolución 186/03 del 09/09/03.

[57] Setubal, J. y Meidanis, J. *Introduction to Computational Molecular Biology* - Univ. de Campinas, Brasil,1997.

[58] Sjolander, K. Karplus, K. Brown, M. Hughey, R. Krogh, A. Mian, S. y Haussler, D. *Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology*. Computer Applications in the Biosciences, 12:327-345, 1996.

[59] Smith y Waterman. Identification for common molecular subsequences. Journal of Molecular Biology 147: 195-197, 1981.

[60] Storno, G y Hartzell, G. *Identifying rpotein binding sites from unaligned DNA fragments*, Proceedings of National Academy of Science of USA 86: 1183-1187, 1989.

[61] Tanner M.A., Wong W.H., *The calculation of posterior distributionsby data augmentation*, J. Am. Stat. Assoc. 1987.

[62] Thompson, J., Higgins, D. y Gibson, T. *Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*. Nucleic Acids Res, 22:4673-4680, 1994.

[63] Walsh, B. , *IntroductiontoBayesianAnalysis* , Lecture Notes for EEB596z, 2002.

[64] Walsh, B. , *Markov Chain Monte Carlo and Gibbs Sampling* , Lecture Notes for EEB596z, 2002.

[65] Witten, I., Frank, E., Trigg, L., Hall, M., Geoffrey, H. y Cunningham, S., *Weka: Practical machine learning tools and techniques wuth java implementations*. Departament of Computer Science. University of Waikato. New Zeland. www.cs.waikato.ac. nz/~ml/publications, 1999.

[66] Witten, I., Frank, E., Trigg, L., Hall, M., y Geoffrey, H. , *Data mining in bioinformatics using Weka*,Bioinformatics, Applications Note, 2004.

[67] Workshop: *Markov Chain Monte Carlo: innovations and applications in statistics, physics and bioinformatics* – National University of Singapure. March 2004.

# Referencias a sitios Web

[ALGGEN]Algorithmics Genetics Group–Cataluña, España-http://www.lsi.upc.es/~alggen/docencia/ember/intro-ember.html
[BLAST] http://www.ncbi.nlm.nih.gov/BLAST/ , http://www.ebi.ac.uk/blastall/index.html, 2004
[BUGS] BayesianinferenceusingGibbsSampling, http://www.mrc-bsu.cam.ac.uk/bugs/Welcome.html
[EBI] http://www.ebi.ac.uk/, 2004
[EMBL] European Bioinformatics Institute, http://www.ebi.ac.uk/index.html, 2004
[EMBnet] http://www.embnet.org/ , 2004
[FASTA] http://fasta.bioch.virginia.edu/ - http://www.ebi.ac.uk/fasta33/index.html
[GenBank] [http://www.ncbi.nlm.nih.gov/Genbank/ - 2004
[GIBBS] http://bayesweb.wadsworth.org/gibbs/gibbs.html – 2004
[GIBBS_SOFT] http://www.fas.harvard.edu/~junliu/Software/gibbs9_95.tar - Diciembre 2003
[HAEMA] http://www.haema.de/ -2004
[IMGT] http://imgt.cines.fr/
[LEPA] http://www.lsi.upc.es/~alggen/docencia/ember/lepa/Tfc1.htm
[LIU] http://www.people.fas.harvard.edu/~junliu/index1.html#Computational_Biology -2002
[MEDLINE] http://www.medline.com/ -2004
[MEGA] http://www.megasoftware.net/index.html
[MEME] http://meme.sdsc.edu /meme/website/ - 2004
[NCBI] National Center for Biotechnology Information . http://www.ncbi.nlm.nih.gov, 2004
[PFAM] http://www.sanger.ac.uk/cgi-bin/Pfam/, http://www.sanger.ac.uk/Software/Pfam/index.shtml- Nov. 2004
[PROSITE] http://www.expasy.ch/prosite/ - 2004
[PIR] Protein Identification Resource, http://pir.georgetown.edu/ , 2004
[SHAMIR] Shamir, Ron, Algorithms in Molecular Biology, http://www.math.tau.ac.il/~rshamir , Tel Aviv University School of Computer Science , 2002
[SwissProt] http://www.ebi.ac.uk/swissprot/, 2004
[WEKA] http://www.cs.waikato.ac.nz/ml/weka/

# Anexo I

## Las proteínas

Las proteínas son polímeros cuyos monómeros se denominan aminoácidos. Una proteína tiene, por lo menos, 100 aminoácidos y como existen 20 aminoácidos distintos, la cantidad de proteínas diferentes que se pueden formar es enorme, y excede cualquier intento de enumeración exhaustiva.

Los aminoácidos se identifican por una letra:

| | |
|---|---|
| A alanina | N asparraguina |
| C cisteína | P prolina |
| D aspartato | Q glutamina |
| E glutamato | R arginina |
| F fenilalanina | S serina |
| G glicina | T treonina |
| H histidina | V valina |
| I isoleucina | W triptófano |
| K lisina | Y tirosina |
| L leucina | X cualquier |
| M metionina | |

La importancia de las proteínas se debe a la variedad de funciones que realizan:

- Forman parte de la estructura del organismo: huesos, tendones, pelo, cáscara, etc.
- Están contenida en la sangre de los vertebrados: hemoglobina.
- Muchas proteínas son hormonas: insulina.
- Anticuerpos: globulina.
- Reserva de alimento: albúmina, caseína (leche materna).
- Movimiento de fibras musculares.
- Enzimas.

Una proteína puede tener, en promedio, unos 200 aminoácidos; aunque proteínas grandes pueden superar los 1000 aminoácidos, un ejemplo de proteína:

**Nitrogen assimilation regulatory protein ntrC - Bradyrhizobium sp**

MPAGSILVADDDTAIRTVLNQALSRAGYEVRLTGNAATLWRWVSQGEGDLVITDVVMPDENAFDLLPRIK
KMRPNLPVIVMSAQNTFMTAIRPSERGAYEYLPKPFDLKELITIVGRALAEPKERVSSPADDGEFDSIPL
VGRSPAMQEIYRVLARLMQTDLTVMISGESGTGKELVARALHDYGRRRNGPFVAVNMAAIPRDLIESELF
GHERGAFTGANTRASGRFEQAEGGTLFLDEIGDMPMEAQTRLLRVLQQGEYTTVGGRTPIKTDVRIVAAS
NKDLRILIQQGLFREDLFFRLNVVPLRVPPLRERIEDLPDLIRHFFSLAEKDGLPPKKLDAQALERLKQH
RWPGNVRELENLARRLAALYPQDVITASVIDGELAPPAVTSGSTATVGVDNLGGAVEAYLSSHFSGFPNG
VPPPGLYHRILKEIEIPLLTAALAATRGNQIRAADLLGLNRNTLRKKIRDLDIQVYRSGG

## Los ácidos nucleicos

Los ácidos nucleicos son polímeros en cuya estructura se repite una unidad llamada nucleótido formando largas cadenas que se enrollan sobre sí mismas. Estas macromoléculas controlan la producción de las proteínas celulares, y además, contienen y transmiten la información hereditaria.

Una célula contiene dos tipos de ácidos nucleicos: el ADN (ácido desoxirribonucleico) y el ARN (ácido ribonucleico), o las siglas en inglés DNA y RNA.

Cada nucleótido contiene fosfato, azúcar, y una de las 4 moléculas base:

| | | | | |
|---|---|---|---|---|
| En el ADN : | Adenina (A) | Timina (T) | Citosina (C) | Guanina (G) |
| En el ARN : | Adenina (A) | Uracilo (U) | Citosina (C) | Guanina (G) |

De esta manera se puede construir diferentes moléculas de ADN o ARN, simplemente variando el número y orden de los nucleótidos que la componen.

La longitud del DNA humano es mayor a $3 \times 10^9$ pares bases. Se denominan "pares bases" pues la estructura del DNA es de doble hélice, un nucleótido en cada hélice. Cada par base consiste en una base purina (A o G) y una pirimidina (C o T) unidos por hidrógeno.