Universidad Zaragoza
1542

Facultad de Ciencias
Universidad Zaragoza

TRABAJO DE FIN DE GRADO EN FÍSICA

DEPARTAMENTO DE FÍSICA TEÓRICA

Técnicas de campo medio generalizado aplicadas a problemas de biofísica

---

Generalised mean field techniques applied to biophysics problems

---

Daniel Ulibarri Sánchez

Tutor:

Pierpaolo Bruscolini

21 de noviembre de 2024

A mis padres y hermana,
por su cariño y apoyo constante.

A mi tutor Pier,
por su paciencia y consejos.

A Michelle,
por ser mi tierra firme.

Índice

Introducción

La aproximación de campo medio es una técnica clásica empleada en el análisis de las propiedades termodinámicas de sistemas físicos macroscópicos, donde muchos componentes interaccionan entre sí, determinando comportamientos colectivos. Estos sistemas de gran número de elementos también aparecen de manera recurrente al investigar sistemas biológicos, lo que hace que sea una técnica muy extendida en el campo de la biofísica. Si bien los resultados que arroja son precisos cuando el tamaño del sistema es muy grande, la aproximación de campo medio suele fallar al tratar con sistemas cerca de sus puntos críticos, y está limitada a sistemas en equilibrio.

Estas limitaciones han impulsado el desarrollo de técnicas de campo medio generalizadas que cubran un mayor espectro de problemas, definidos en redes no necesariamente regulares. La idea general de estos enfoques es definir unas regiones de la red (clústeres) cuyos estados se tratan de manera exacta, mientras que para el resto de la red se emplea la aproximación de campo medio. En este proceso, resulta crucial controlar cuidadosamente los efectos de la separación en regiones sobre la estimación de la entropía.

En este contexto surgen los algoritmos de propagación de creencias y sus generalizaciones, cuya importancia en el análisis de sistemas físicos definidos en redes con variables discretas ha aumentado en gran medida en los últimos tiempos. Estos algoritmos se basan en la transmisión de información entre nodos de manera local, de forma que los nodos cercanos se van agrupando formando pseudonodos que contienen toda la información del conjunto, con lo que se reduce la dimensionalidad de la red. Se puede demostrar que los puntos fijos de estos algoritmos se corresponden con los mínimos de una función energía libre definida sobre el sistema [1], con lo que el problema de encontrar estados de equilibrio se puede transformar en un problema de minimización bajo restricciones, resoluble mediante métodos variacionales.

Las generalizaciones de los métodos de propagación de creencias se centran, por un lado, en mejorar las propiedades de convergencia de los métodos, y por otro, en ampliar su aplicación a problemas de dinámica, y no solo de equilibrio. En esta línea, surgen artículos como el de Pelizzola y Pretti [2], que buscan construir un método aproximado para la simulación de la dinámica en redes, haciendo uso de los fundamentos del método de variación en clúster.

En este trabajo, que retoma y desarrolla una investigación previa [3, 4], se va a emplear el método propuesto en el artículo de Pelizzola y Pretti para aplicarlo a distintos sistemas que simulen una red de regulación genética, analizando los resultados que arroja y comparándolos con los correspondientes a otros métodos clásicos de estudio de dinámica en redes.

Estructura de la memoria

La memoria se estructura de la siguiente manera. En la sección 1 se introducen los conceptos de red de regulación genética y de red booleana y se explican sus distintas representaciones y los tipos de evolución que se pueden definir sobre ellas.

La sección 2 se dedica a presentar el método de variación de clúster, tanto en su versión clásica como en su versión dinámica, y los dos otros métodos que se van a emplear en el análisis de resultados (método de Monte Carlo y método de la matriz de transición), así como los modelos sobre los que se van a aplicar los métodos (modelo de juguete de tres nodos y modelo de ciclo celular).

En la sección 3 se recogen los resultados de aplicar los distintos métodos sobre los modelos presentados en la sección 2. Estos dos modelos se habían considerado ya en una investigación anterior [3, 4], centrando el análisis en la evolución temporal a partir de algunas condiciones iniciales, utilizando el programa *MaBoSS* [5] como referencia. Aquí, después de remodelar en profundidad el código previo para poder usarlo sobre una red cualquiera, de cara a poder estudiar perturbaciones y modificaciones en las uniones de las redes, se utilizan esos modelos para profundizar en el estudio del comportamiento del algoritmo. En particular, se muestran los resultados de una exploración sistemática de las condiciones iniciales de la dinámica, que reveló una inestabilidad del algoritmo de Pelizzola y Pretti, diseñado inicialmente para grafos no dirigidos, cuando se aplica a las redes de regulación genética. A continuación, se explica la estrategia adoptada para hacer frente a este problema, y los resultados que se obtienen.

Por último, la sección 4 se destina para las conclusiones del trabajo, así como posibles líneas de investigación futuras relacionadas con el tema. Al final del documento se incluyen una serie de apéndices que extienden y complementan el contenido del texto principal.

1. Marco teórico

La complejidad de los sistemas biológicos propone retos importantes a la hora de formular modelos para su descripción, y el paradigma de la redes complejas representa un marco adecuado, aunque simplificado, para esa tarea. Por lo tanto, podemos hablar de redes biológicas para describir sistemas completamente distintos, abarcando escalas tan dispares como las relaciones interespecíficas de un ecosistema o los sistemas de regulación genética en el interior de las células.

De manera general, las redes biológicas se pueden definir como un sistema biológico organizado compuesto de unidades que interactúan entre sí de acuerdo a reglas regulatorias con el fin de llevar a cabo una función específica [6]. Dichas unidades se representan con los nodos de la red, mientras que los enlaces representan las relaciones de interación, regulación, etc. Los bloques constituyentes de las redes biológicas pueden oscilar en complejidad desde biomoléculas hasta organismos completos.

El vasto espectro de campos de investigación que abarcan hace que se hayan desarrollado diferentes métodos de estudio adaptados a las características concretas de cada sistema.

Una de las clasificaciones de las redes biológicas consiste en distinguir el espacio de posibles estados en los que se puede encontrar un nodo. Según esta, se puede hablar de espacios de estados continuos ($\subset \mathbb{R}$) o discretos ($\subset \mathbb{N}$). Dentro de los discretos, hay una familia de redes cuyos nodos solo toman dos valores, que se suelen representar por 1 (activo) y 0 (inactivo). A esta clase pertenecen las redes booleanas que son las estructuras matemáticas subyacentes a las redes de regulación genética que se van a modelizar en este trabajo.

1.1. Redes de regulación genética

Las redes de regulación genética son un tipo particular de redes biológicas, mediante las que se modeliza la respuesta de una célula a su entorno, así como la regulación de todos los procesos metabólicos, de señalización y de diferenciación celular. En estas redes los nodos pueden representar los tres actores relevantes en la regulación: ADN (genes); ARN mensajeros, obtenidos de la transcripción de los genes; y proteínas, obtenidas de la traducción del ARN, y que a su vez actúan como reguladoras de los genes. Sin embargo, también es usual agrupar un gen con su ARN y su proteína, describiéndolos como un único nodo. Esto implica renunciar a la descripción precisa de las dinámicas y de las escalas temporales de la transcripción y la traducción.

Aunque se suelen describir preferentemente con sistemas de ecuaciones diferenciales, las redes de regulación genética son también un dominio de aplicación habitual de modelos booleanos, que por su sencillez y ausencia de parámetros, representan una buena herramienta para averiguar cualitativamente el comportamiento de redes grandes. Así, los modelos booleanos son actualmente la construcción matemática empleada para modelizar una gran variedad de mecanismos moleculares de regulación [7].

1.2. Redes genéticas booleanas

En una red genética booleana, las aristas (o enlaces) son dirigidas, representando acciones de activación o inhibición. Varias aristas, procedentes de distintos nodos $i_1, \dots i_k$, pueden incidir sobre el mismo nodo j, de forma que el estado de este último evoluciona en función de los valores de aquellos, combinándolos según una expresión booleana. Conocer la expresión explícita de esta función booleana para cada nodo es fundamental para caracterizar completamente la red, y es

una de las tareas más complicadas en la práctica, siendo que muchas veces la relaciones entre genes no se conocen de antemano, y se infieren de los datos experimentales.

Por otro lado, con k nodos incidentes, hay 2^k valores de entrada diferentes, y 2^{2^k} posibles funciones binarias asociadas, así que una elección de la función de un nodo basada en pruebas sistemáticas es inabordable en la práctica. Afortunadamente, suelen observarse reglas razonablemente sencillas de combinación. De esta manera, dado un nodo con varios posibles activadores, se puede comúnmente asumir que todos sus activadores tienen una contribución igual, con lo que se combinan mediante el operador lógico "OR" ($\vee$). Recíprocamente, si se tiene un nodo que puede ser inhibido por varias substancias, todos sus inhibidores se combinan mediante el operador lógico "AND" ($\wedge$). Así, un nodo solo estará activo cuando al menos uno de sus activadores esté presente, y ninguno de sus inhibidores esté activo. [8].

Los valores concretos de sus nodos evolucionan con el tiempo, de acuerdo con la expresión de la función lógica que combina los estados de los nodos vecinos. A la hora de actualizar los valores de los nodos, se pueden seguir tres estrategias principales: síncrona, asíncrona y probabilística.

Cuando se emplean actualizaciones síncronas, el estado de todos los nodos se actualiza (de acuerdo a unas reglas fijas) simultáneamente en cada paso temporal, con lo que la transición concreta que tiene lugar depende en su totalidad del estado inicial. De esta manera, se obtiene una dinámica determinista muy robusta.

Por su parte, cuando se permiten actualizaciones asíncronas, en cada paso temporal se considera tan solo la posible transición de uno de los nodos (elegido al azar). Esto provoca que se tengan N posibles transiciones (siendo N el número de nodos del sistema), lo que se asemeja más al comportamiento real de los sistemas biológicos.

Por último, al considerar actualizaciones probabilísticas, las reglas que dictan la evolución entre dos estados consecutivos se expresan en términos de probabilidades de transición. De esta manera, el sistema se puede expresar como una cadena de Markov (ver apéndice A). Es este último tipo de evolución el que se va a emplear en este trabajo.

Como regla general, el conjunto de nodos de la red se representará mediante $\{X_1, X_2, \ldots, X_N\}$, y cada nodo se corresponderá con una variable aleatoria que toma valores en $\{0, 1\}$. Los valores concretos que toma cada nodo se denotarán como x_i. Dado que se ha considerado una evolución estocástica, la solución a los problemas de inferencia consistirá en obtener la distribución de probabilidad conjunta $p\left(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\right)$, que denotaremos $p(\mathbf{x})$, para un tiempo dado. En ocasiones, tan solo será de interés la distribución de probabilidad $p\left(\mathbf{x}_a\right)$ de algún subconjunto $\{X_a\} \subset \{X_1, X_2, \ldots, X_N\}$, al que llamaremos clúster.

1.3. Representación de una red booleana

Para el estudio de las redes en general, y de las booleanas en particular, se han desarrollado numerosas representaciones gráficas que facilitan tanto la visualización de la estructura de las redes como el cálculo de trayectorias sobre las mismas.

Entre las representaciones gráficas más comunes, se encuentran las redes bayesianas, los campos aleatorios de Markov por pares y los grafos de factores. Cada representación tiene unas características propias que la favorecen a la hora de visualizar clases de problemas diferentes, si bien todas son equivalentes como se demuestra en [9]. Un ejemplo de cada tipo de representación se puede encontrar en la figura 1.

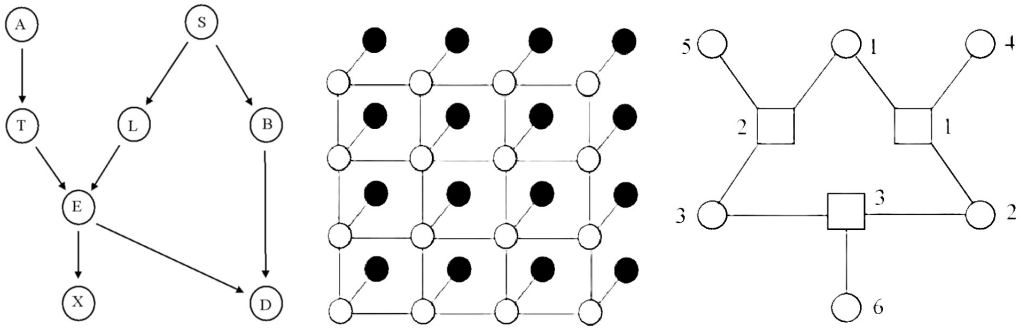

Figura 1: Ejemplos de las distintas representaciones de una red booleana. A la izquierda, una red bayesiana representando un problema de inferencia para el diagnóstico de una enfermedad. En el centro, un campo aleatorio de Markov por pares representando un problema de lectura de píxeles. A la derecha, un grafo de factores empleado para la resolución de problemas de códigos de control de paridad. Imágenes obtenidas de [9].

1.3.1. Redes bayesianas

La representación como red bayesiana consiste en mostrar en un grafo dirigido acíclico todos los nodos del sistema, así como las conexiones entre ellos, entendiendo que existe una conexión entre el nodo A y el nodo B si el nodo A afecta al estado del nodo B en el siguiente paso temporal (notar que B no tiene por qué afectar a A). Se dirá entonces que A es un padre de B. Cabe destacar que esta representación es adecuada para casos en los que la red sea un grafo dirigido acíclico, o en los que los ciclos existentes se puedan alterar (ya sea agrupando nodos o reduciendo ciclos de tamaño mayor que la longitud de correlación) para conseguir la estructura de grafo buscada.

Al trabajar con redes booleanas estocásticas se tiene que cada conexión entre nodos se corresponde con una probabilidad condicionada. De esta manera, la probabilidad de que un nodo se encuentre en uno de sus estados depende solo directamente del estado de sus padres. Para los nodos que no tengan ningún padre se asumirá que sus distribuciones de probabilidad son independientes. Así, la distribución de probabilidad conjunta se obtendrá como

$$p\left(\mathbf{x}\right) = \prod_{1}^{N} p\left(x_i \mid \mathrm{Par}\left(\mathrm{x_i}\right)\right), \tag{1.1}$$

donde $\mathrm{Par}\left(\mathrm{x_i}\right)$ es el estado de los padres del nodo X_i ($p\left(x_i \mid \mathrm{Par}\left(\mathrm{x_i}\right)\right) = p\left(x_i\right)$ en caso de que no tenga padres).

Para obtener la distribución marginal correspondiente a un clúster $p\left(\mathbf{x}_a\right)$ hay que sumar sobre todos los posibles estados de los nodos que no se encuentren en dicho clúster

$$p\left(\mathbf{x}_a\right) = \sum_{i|X_i \notin X_a} \sum_{x_i} p\left(x_1, x_2, \ldots, x_N\right), \tag{1.2}$$

lo que hace que el coste computacional crezca exponencialmente con el tamaño del clúster. Este hecho será de gran importancia a la hora de buscar métodos de simulación, ya que limitará en gran medida el tamaño de las redes para las que se pueden calcular probabilidades marginales de manera directa.

1.3.2. Campos aleatorios de Markov por pares

En los campos aleatorios de Markov por pares se trabaja con dos conjunto diferentes de nodos $\{X_1, X_2, \ldots, X_N\}$ y $\{Y_1, Y_2, \ldots, Y_N\}$. Las variables Y_i toman valores $\{y_i\}$ y se pueden medir directamente, mientras que las variables X_i, que toman valores en $\{x_i\}$, son magnitudes subyacentes de las que se quiere obtener información. Entre x_i e y_i existe una dependencia estadística, denominada evidencia de x_i que se escribe como $\phi_i(x_i, y_i)$.

Este tipo de representación se suele emplear para el reconocimiento de imágenes en problemas de visión artificial, por lo que se asume cierta estructura sobre las variables X_i subyacentes. Para ello, se introduce una función de compatibilidad entre los valores de la variable X_i y los valores de sus vecinos (en un sentido tan laxo como se necesite) $\{X_j\}$ que se suele expresar como $\psi_{ij}(x_i, x_j)$.

En este caso, la distribución de probabilidad conjunta debe incluir tanto las variables observables como las subyacentes, con lo que tiene la forma

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{\{i \leftrightarrow j\}} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i, y_i), \tag{1.3}$$

donde Z es una constante de normalización y $\{i \leftrightarrow j\}$ representa el conjunto de todos los pares de vecinos posibles.

El cálculo de probabilidades marginales se haría de manera análoga a las redes bayesianas, y conlleva igualmente el problema del crecimiento exponencial en el coste computacional. Sin embargo, a diferencia de las redes bayesianas, esta representación se puede emplear de manera directa para grafos no dirigidos, puesto que en lugar de emplear probabilidades condicionales entre las variables $\{X_i\}$ las relaciones entre estas se codifican en las funciones de compatibilidad $\{\psi_{ij}\}$.

1.3.3. Grafos de factores

Partimos del conjunto de nodos $\{X_1, X_2, \ldots, X_N\}$ que toman valores $\{x_i\}$. De manera general, $p(\mathbf{x})$ se puede escribir como

$$p(\mathbf{x}) = \frac{1}{Z} \prod_\alpha \Psi_\alpha(\mathbf{x}_\alpha), \tag{1.4}$$

donde Z es una constante de normalización y las funciones Ψ_α son M funciones indexadas por un parámetro α que toman como argumento algún subclúster $(\mathbf{X}_\alpha)$.

Un grafo de factores es un grafo bipartito que codifica la estructura de factorización de las probabilidades en la ecuación (1.4). En este caso, los dos conjuntos disjuntos que forman el grafo bipartito serían los valores de los nodos $\{x_1, x_2, \ldots, x_N\}$, que se representan por un círculo, y las funciones $\{\Psi_A, \Psi_B, \ldots, \Psi_M\}$, que se representan con un cuadrado. Las únicas conexiones permitidas son entre valores de nodos y funciones, y se considerará que existe una conexión entre x_i y Ψ_α cuando x_i sea un argumento de la función Ψ_α.

Los grafos de factores han sido ampliamente empleados en la resolución de códigos de control de paridad (códigos que controlan la correcta transmisión de mensajes mediante el uso de bits de paridad), si bien su campo de aplicación es mucho más extenso como se verá a continuación. De nuevo, a la hora de calcular probabilidades marginales se encuentran problemas con el coste computacional.

2. Metodología

El principal objetivo de este trabajo es analizar la evolución temporal de redes booleanas. Para ello, se van a emplear el método de variación de clúster, el método de Monte Carlo y el método de la matriz de transición. Todos ellos van a ser implementados computacionalmente de forma que se puedan simular distintos modelos. Inicialmente, se van a aplicar los métodos sobre un modelo de juguete de tres nodos suficientemente pequeño como para poder seguir la evolución de los estados, pero con la complejidad necesaria para presentar estados y ciclos atractores. Tras esto, se aplicarán sobre un modelo de diez nodos más complejo que imita el ciclo celular.

2.1. Método de variación en clúster (CVM)

El método de variación en clúster, CVM por sus siglas en inglés (*Cluster Variation Method*), es una jerarquía de técnicas variacionales apoximadas para realizar estadística inferencial sobre modelos discretos en equilibrio [10]. Históricamente, el método se ha empleado para determinar diagramas de fase para transiciones de primer y segundo orden con esfuerzos computacionales moderados y con resultados comparables a los obtenidos mediante simulaciones de Monte Carlo [11]. El método fue propuesto por Kikuchi (1951) [12], pero por su mayor simplicidad, se va a seguir aquí la presentación del mismo por parte de Heskes y colaboradores [13].

Se parte de la distribución de probabilidad presentada al hablar de los grafos de factores en la ecuación 1.4. Entendiendo las funciones Ψ_α como potenciales, se puede establecer un paralelismo con el formalismo canónico de la física estadística, donde la energía asociada a cada clúster vendrá dada por $\psi_\alpha(\mathbf{x}_\alpha) = \log \Psi_\alpha(\mathbf{x}_\alpha)$. Notar que, para un sistema no físico, la Ley de Boltzmann

$$p(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})/k_B T} \tag{2.1}$$

se puede entender como un postulado que define la energía del sistema [1]. En ese caso, tanto la temperatura (T) como la constante de Boltzmann (k_B) se pueden elegir de manera arbitraria puesto que solo determinan la escala para las unidades en la que se mide la energía, con lo que se puede tomar $k_B T = 1$.

En general, para calcular la constante de normalización y las probabilidades condicionales sobre los clústeres hay que sumar sobre el número de estados, que crece de manera exponencial tanto con el tamaño de la red como con el tamaño de los clústeres. Mediante esta distribución de probabilidad, se pueden obtener los valores de la energía (E), la entropía (S), la energía libre de Helmholtz (F) y la constante de normalización (Z) según las conocidas relaciones termodinámicas

$$E(p) = -\sum_\alpha \sum_{\mathbf{x}_\alpha} p(\mathbf{x}_\alpha)\, \psi_\alpha, \tag{2.2}$$

$$S(p) = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) \tag{2.3}$$

y

$$F(p) = E(p) - S(p) = -\log Z. \tag{2.4}$$

La idea del CVM consiste en evitar la suma exponencial sobre el número de estados del sistema, calculando en su lugar una aproximación de F. Tras ello, se buscará minimizar $F(p)$ sobre el conjunto de las distribuciones de probabilidad sobre el sistema ($p(\mathbf{x})$). Para realizar esta

aproximación se expresará $p(\mathbf{x})$ a través de las distribuciones de probabilidad definidas sobre los clústeres (probabilidades marginales), que a priori no tienen por qué ser disjuntos.

Existe una selección mínima de clústeres que satisfacen la factorización de los potenciales en la ecuación 1.4, a los que llamaremos clústeres maximales. Notar que esta elección distingue entre las diferentes realizaciones del método y determinará el balance entre precisión y complejidad computacional. Cuanto más grandes sean los clústeres maximales, más precisa será la aproximación, pero la complejidad de cálculo será mayor, puesto que esta crece de manera exponencial con su tamaño. Como ejemplo de elecciones de estos clústeres maximales, cuando se elige que cada clúster solo contenga un nodo, se recupera la apoximación de campo medio, y cuando cada clúster contiene un nodo y todos sus primeros vecinos, se recupera la aproximación de Bethe-Peierls [2, 14].

Volviendo al caso general, en la aproximación de la energía libre mediante el CVM se deja el término asociado a la energía sin modificar, y se busca aproximar la entropía mediante una suma de las entropías marginales de forma que

$$F(p) \approx F_{CVM}(p) = E(p) - S_{CVM}(p), \tag{2.5}$$

donde S_{CVM} toma la forma

$$S_{CVM}(p) = \sum_{\alpha \in \{maxClust\}} S_\alpha(p) + \sum_{\beta \in \{subClust\}} c_\beta S_\beta(p), \tag{2.6}$$

donde $S_{\alpha,\beta}$ es la entropía marginalizada a un clúster (calculada restringiendo el sumatorio de la ecuación 2.3 a los posibles estados del clúster) y los c_β son los números de Möbius o de sobreconteo, cuyo papel se explica a continuación.

En la ecuación anterior, el primer término suma las entropías de los clústeres maximales. No obstante, dado que los clústeres maximales no son disjuntos, al sumar sus entropías se está sobrecontando la contribución a la entropía total de algunos de los nodos. Para corregir esto, se añade el segundo sumando, en el que el sumatorio ya no se efectúa sobre los clústeres maximales, sino que recorre los subclústeres obtenidos mediante intersecciones de clústeres maximales, interesecciones de intersecciones, y cualquier otra intersección sucesiva. Así se añade la entropía asociada a estos subclústeres modulada por los coeficientes c_β de forma que la contribución a la entropía de cada nodo solo aparezca una vez.

Notar que si bien la ecuación 2.6 se puede obtener de manera directa razonando sobre la aproximación del sistema como un conjunto de clústeres, esta también se puede deducir de manera formal como se recoge en el apéndice B. Como se demuestra allí, lo que realmente se está realizando es el truncamiento de la expansión de S como serie de cumulantes.

Volviendo a la ecuación 2.6, de la idea de que los coeficientes c_β evitan el sobreconteo de algunos nodos se puede deducir que

$$c_\beta = 1 \quad \forall \beta \in U, \quad c_\beta = 1 - \sum_{\alpha \supset \beta} c_\alpha \forall \quad \beta \in V \tag{2.7}$$

donde U es el conjunto de los clústeres maximales, y V el conjunto de todos sus subclústeres.

Cabe destacar, por último, que por la aproximación que se ha realizado de la entropía, la aproximación del CVM de la energía libre solo depende de las probabilidades marginales definidas sobre los clústeres. Sustituimos así la minimización de la energía libre sobre la distribución de

probabilidad conjunta $p(\mathbf{x})$ por la minimización de la energía libre definida en la ecuación 2.5 sobre un conjunto de pseudomarginales $Q = \{Q_\alpha\}$ consistentes y normalizadas de acuerdo a las ecuaciones

$$\sum_{x_{\gamma' \setminus \gamma}} Q_{\gamma'}\left(x_{\gamma'}\right) = Q_\gamma\left(x_\gamma\right) \quad \forall \gamma' \supset \gamma \tag{2.8}$$

y

$$\sum_{x_\gamma} Q_\gamma\left(x_\gamma\right) = 1 \quad \forall \gamma. \tag{2.9}$$

Lo que se espera del método CVM es que estas pseudomarginales sean aproximaciones precisas de las marginales exactas $p(\mathbf{x}_\alpha)$. La entropía calculada usando las distribuciones pseudomarginales será exacta siempre que el grafo de región asociado al sistema sea simplemente conexo.

2.1.1. Método de variación de clúster cinético

Como se ha comentado antes, el CVM clásico se emplea para analizar sistemas en equilibrio. No obstante, en los últimos años se han llevado a cabo cada vez más estudios orientados a investigar la dinámica en redes complejas, con especial interés en el papel que juega la red [2]. Así, diferentes adaptaciones del CVM en las que las probabilidades de transición entre estados juegan el papel de interacciones han cobrado mayor importancia a la hora de realizar estadística inferencial sobre sistemas que evolucionan con el tiempo. En este caso, se va a emplear la llamada aproximación PQR, extraída del artículo de Pelizzola y Pretti [2].

Partimos de una red booleana que evoluciona con el tiempo de manera estocástica. Como se ha indicado, esta situación es equivalente a contar con una cadena de Markov en la que cada nodo tiene asociado una variable aleatoria, cuyo valor cambia con el tiempo $\{X_i^{(t)}\}_{i=1,\ldots,N}^{t=0,\ldots,\tau}$. Una trayectoria particular de la cadena de Markov sería $\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(\tau)}$, y tendría asociada una probabilidad

$$p\left(\mathbf{X}^{(0)} = \mathbf{x}^{(0)}, \ldots, \mathbf{X}^{(\tau)} = \mathbf{x}^{(\tau)}\right) = p^{(0)}\left(\mathbf{x}^{(0)}\right) \prod_{t=0}^{\tau-1} w^{(t)}\left(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{(t)}\right), \tag{2.10}$$

donde se ha usado la hipótesis de una dinámica de Markov, y donde $p^{(0)}\left(\mathbf{x}^{(0)}\right)$ es la probabilidad inicial del estado $\mathbf{x}^{(0)}$ y $w^{(t)}\left(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{(t)}\right)$ la probabilidad de transición entre $\mathbf{x}^{(t)}$ y $\mathbf{x}^{(t+1)}$.

Restringiendo que el estado de cada nodo en el siguiente paso temporal solo dependa de su estado actual y el de sus vecinos (lo cual es válido en la mayoría de sistemas físicos), la probabilidad de transición factoriza como

$$w^{(t)}(\mathbf{y} \mid \mathbf{x}) = \prod_i w_i{}^{(t)}\left(y_i \mid x_{i,\partial i}\right), \tag{2.11}$$

donde $\mathbf{x}$ e $\mathbf{y}$ se corresponden con dos configuraciones cualesquiera de la red, x_i e y_i serían las configuraciones de cada nodo, y $x_{i,\partial i}$ una abreviación de la configuración de x_i y de todos sus vecinos. Cabe destacar que para sistemas arbitrarios este no tiene por qué ser el caso, con lo que el razonamiento siguiente dejaría de ser válido.

Consideremos ahora la entropía asociada a la distribución sobre las trayectorias de la cadena

$$S(p) = -\sum_{\mathbf{x}^{(0)},\ldots,\mathbf{x}^{(\tau)}} p\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(\tau)}\right) \ln p\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(\tau)}\right). \tag{2.12}$$

Definiendo una función potencial como

$$\varepsilon\left(\mathbf{x}^{(0)},\ldots,\mathbf{x}^{(\tau)}\right) = -\ln p^{(0)}\left(x^{(0)}\right) - \sum_{t=0}^{\tau-1} \ln w^{(t)}\left(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{(t)}\right), \tag{2.13}$$

se puede construir un análogo de energía libre de Helmholtz como

$$F(p) = \sum_{\mathbf{x}^{(0)},\ldots,\mathbf{x}^{(\tau)}} p\left(\mathbf{x}^{(0)},\ldots,\mathbf{x}^{(\tau)}\right)\varepsilon\left(\mathbf{x}^{(0)},\ldots,\mathbf{x}^{(\tau)}\right) - S(p). \tag{2.14}$$

Esta función F sería la energía libre asociada a un sistema en el que el tiempo fuera una dimensión extra. De hecho, se podría visualizar el sistema como una pila de $\tau+1$ copias de la red (indexadas por el parámetro t), con lo que se tendría una red en el espacio-tiempo (notar que las copias de niveles sucesivos están conectadas mediante las probabilidades de transición).

Sobre las probabilidades de transición se ha asumido un carácter local, y la propiedad de Markov dota a la dimensión del tiempo de esa misma localidad, con lo que cabe esperar que la entropía se pueda aproximar de manera bastante precisa mediante un truncamiento de su expansión como serie de cumulantes. Así, se buscará aplicar una generalización del CVM sobre el sistema espacio-temporal construido.

Como se mencionó al introducir el CVM, la elección más relevante que se debe hacer y que establece el balance entre precisión y complejidad es la de los clústeres maximales. Esta elección viene influenciada por el tipo de correlación que se quiera mantener sobre las variables. En un sistema en el que los nodos puedan estar conectados a los vecinos de su capa temporal y posiblemente a los vecinos de la capa temporal posterior (no existen conexiones con más capas temporales por la propiedad de Markov) los posibles clústeres en torno a un nodo vienen dados por la tabla 1.

En este caso, los clústeres maximales elegidos son los de tipo P, Q y R. El tipo P es necesario puesto que incluye toda la información necesaria para calcular las probabilidades de transición según la ecuación 2.11. Por su parte, el clúster Q añade correlaciones temporales entre pares de primeros vecinos, y el clúster R, correlaciones entre un nodo en un paso temporal y sus vecinos en el paso consecutivo. La elección de estos tres clústeres es la llamada aproximación PQR.

La entropía asociada al CVM para la aproximación PQR se deduce de combinar las ecuaciones 2.6 y 2.12, lo que da lugar a

$$\begin{aligned} S_{PQR} = \sum_{t=0}^{\tau-1}\Bigg\{ & \sum_i \left[S\left(P_i^{(t)}\right) + S\left(R_i^{(t)}\right) + (d_i-1)\,S\left(V_i^{(t)}\right)\right] + \sum_{ij} S\left(Q_{ij}^{(t)}\right) \\ & - \sum_{i,j\in\partial i}\left[S\left(T_{i,ij}^{(t)}\right) + S\left(U_{ij,i}^{(t)}\right)\right]\Bigg\} - \sum_{t=1}^{\tau-1}\left\{\sum_i S\left(S_i^{(t)}\right) - \sum_{ij} S\left(Z_{ij}^{(t)}\right)\right\}. \end{aligned} \tag{2.15}$$

Cabe destacar, no obstante, que la entropía S_{PQR} está asociada a toda la trayectoria temporal del sistema. Para poder comparar con otros métodos, se necesita la entropía del sistema en cada paso temporal para lo que hay que marginalizar los clústeres de forma que queden sobre una misma capa temporal. En este caso, se tendrá que

$$S_{PQR}^{(t)} = \sum_i S\left(S_i^{(t)}\right) - \sum_{ij} S\left(Z_{ij}^{(t)}\right). \tag{2.16}$$

	t	$t+1$	Representación	Definición
M	$i, \partial i$	$i, \partial i$	$M_i^{(t)}\left(y_{i,\partial i}, x_{i,\partial i}\right)$	$p\left(X_{i,\partial i}^{(t+1)} = y_{i,\partial i}, X_{i,\partial i}^{(t)} = x_{i,\partial i}\right)$
P	$i, \partial i$	i	$P_i^{(t)}\left(y_i, x_{i,\partial i}\right)$	$p\left(X_i^{(t+1)} = y_i, X_{i,\partial i}^{(t)} = x_{i,\partial i}\right)$
Q	i, j	i, j	$Q_{ij}^{(t)}\left(y_{i,j}, x_{i,j}\right)$	$p\left(X_{i,j}^{(t+1)} = y_{i,j}, X_{i,j}^{(t)} = x_{i,j}\right)$
R	i	$i, \partial i$	$R_i^{(t)}\left(y_{i,\partial i}, x_i\right)$	$p\left(X_{i,\partial i}^{(t+1)} = y_{i,\partial i}, X_i^{(t)} = x_i\right)$
S	$i, \partial i$	–	$S_i^{(t)}\left(x_{i,\partial i}\right)$	$p\left(X_{i,\partial i}^{(t)} = x_{i,\partial i}\right)$
T	i, j	i	$T_{i,ij}^{(t)}\left(y_i, x_{i,j}\right)$	$p\left(X_i^{(t+1)} = y_i, X_{i,j}^{(t)} = x_{i,j}\right)$
U	i	i, j	$U_{ij,i}^{(t)}\left(y_{i,j}, x_i\right)$	$p\left(X_{i,j}^{(t+1)} = y_{i,j}, X_i^{(t)} = x_i\right)$
V	i	i	$V_i^{(t)}\left(y_i, x_i\right)$	$p\left(X_i^{(t+1)} = y_i, X_i^{(t)} = x_i\right)$
Z	i, j	-	$Z_{ij}^{(t)}\left(x_{i,j}\right)$	$p\left(X_{i,j}{}^{(t)} = x_{i,j}\right)$
A	i	–	$A_i^{(t)}\left(x_i\right)$	$p\left(X_i^{(t)} = x_i\right)$

Tabla 1: Resumen de los posibles clústeres existentes en torno a un nodo. En la primera columna el nombre del clúster, en la segunda los nodos que incluye en la capa temporal actual, y en la tercera los nodos de la capa temporal posterior. Notar que i, j hace referencia a un par de primeros vecinos. En la cuarta y quinta columna se recogen la representación que se usará del clúster y su definición en cuanto a variables aleatorias.

Recordemos que la idea de aproximar la entropía del sistema era poder resolver el problema de minimización de la energía libre de Helmholtz, para lo que basta minimizar la entropía en 2.15, lo que resulta un problema complicado debido a la superposición de los clústeres. En general, esta minimización se puede realizar de manera iterativa mediante técnicas de propagación de creencias, que no aseguran la convergencia, o con otros métodos más complejos. No obstante, en el caso de un sistema espacio-temporal proveniente de una cadena de Markov como el que se ha planteado, existe un procedimiento mucho más sencillo, como se recoge en el apéndice C.

2.2. Método de Monte Carlo

El método de Monte Carlo es un método clásico en el estudio de modelos estocásticos, puesto que permite estimar las probabilidades de los distintos estados en sistemas en los que la evolución exacta sea difícil de calcular. La idea del método es simular un gran número de veces la evolución del sistema para aproximar algún estimador realizando estadística sobre los resultados obtenidos.

Al trabajar con cadenas de Markov, lo que se busca generar es un gran número de trayectorias independientes partiendo de unas mismas condiciones iniciales. En cada trayectoria, existe una cierta probabilidad de transición (que en este caso vendrá dada por las w_i de la ecuación 2.11) entre el estado de un nodo a tiempo t y los posibles estados del nodo accesibles para el paso $t+1$. Así, en función de dicha probabilidad se decidirá si se acepta o no la transición, repitiendo el proceso para todos los pasos. En concreto, la implementación utilizada sigue un proceso de actualización de los nodos síncrono. De esta manera, en cada paso temporal, la probabilidad de

transición de cada uno de los nodos se evalúa de manera independiente al resto, dependiendo tan solo del estado del sistema a tiempo t. Esto puede dar lugar a los dos casos extremos de que el sistema no modifique el estado de ninguno de sus nodos, o de que modifique el estado de todos.

La clave del método consiste en que, al promediar los resultados de una magnitud para todas las trayectorias, se obtiene un estimador de la misma. Las leyes de los grandes números aseguran la convergencia a su valor esperado del estimador correspondiente a n trayectorias, si bien el error cometido tan solo decrece como $1/\sqrt{n}$. Esto hace que se necesiten muestras muy grandes para obtener valores precisos. No obstante, este error se puede reducir en gran medida empleando técnicas de reducción de varianza como se recoge en [15].

2.3. Evolución mediante matriz estocástica

Este método de cálculo no supone sino una aplicación directa de las propiedades de las cadenas de Markov en tiempo discreto homogéneas. Partiendo de la ecuación A.1 es fácil ver que la distribución de probabilidad en un instante t es el resultado de aplicar la matriz de transición t veces sobre la distribución inicial. Usando la nomenclatura introducida al presentar el CVM, las entradas de dicha matriz de transición se corresponderían con las $w(\mathbf{y} \mid \mathbf{x})$ de la ecuación 2.11. Bajo la hipótesis de factorización de las probabilidades se podrán entonces relacionar las probabilidades de transición de cada nodo con las probabilidades de transición de los estados completos de la red. Notar asimismo que, en ese caso, la distribución inicial debe hacer referencia a la red completa, con lo que en caso de contar con las condiciones iniciales de los nodos individuales, estas se deben combinar en una probabilidad global.

De los métodos presentados aquí, es el único que obtiene la distribución de probabilidad exacta para cada paso temporal, si bien su coste computacional crece exponencialmente con el tamaño del sistema. Esto es fácil de ver notando que los vectores que expresan la probabilidad de cada estado $\mathbf{x}_\alpha$ de la red, $\mathbf{q}(t) = \{p(\mathbf{x}_\alpha, t), \alpha = 1, \ldots, K\}$, tienen longitud $K = 2^N$ siendo N el número de nodos de la red. Así, el número de operaciones a realizar para obtener cada nuevo vector crece de manera exponencial con el tamaño de la red, haciendo que el método sea computacionalmente inservible para sistemas suficientemente grandes. No obstante, para los modelos pequeños que se van a tratar aquí (a lo sumo de diez nodos) servirá para obtener resultados contra los que comparar el CVM.

2.4. Modelos

Como se ha comentado, los modelos cuyo comportamiento se va a estudiar en primer lugar son un modelo de juguete de tres nodos y un modelo del ciclo celular. Es importante destacar la forma en la que se va a modelizar la evolución estocástica, para lo que se va a seguir el enfoque de G. Stoll y colaboradores en [5]. Es de este mismo trabajo del que se han obtenido inicialmente los modelos de estudio.

En dicho artículo, se trabaja con ritmos de transición para cada nodo, en lugar de probabilidades de transición. Esto permite trabajar con sistemas a tiempo continuo, y además proporciona una manera de introducir información biológica cuantitativa para determinadas redes (por ejemplo, los ritmos de decaimiento o de generación de proteínas se pueden medir experimentalmente). Más aún, las probabilidades de transición se pueden recuperar a posteriori a partir de los ritmos definidos, como se explicará a continuación. La elección de este artículo viene motivada por el hecho de que es uno de los pocos trabajos en los que se presenta una descripción probabilísti-

ca de las reglas de evolución para los distintos nodos. Así, el esfuerzo de traducir los ritmos a probabilidades y de pasar de tiempo continuo a discreto se compensa con el hecho de tener un marco de trabajo sobre el que definir modelos booleanos estocásticos arbitratios.

Para un nodo I cualquiera, se tienen definidos un ritmo de activación ρ_{act} y un ritmo de decaimiento ρ_{dec} como

$$\begin{cases} \rho_{act} = \rho_{act_1} \text{ si } cond_1,\ \rho_{act_2} \text{ en otro caso} \\ \rho_{dec} = \rho_{dec_1} \text{ si } cond_2,\ \rho_{dec_2} \text{ en otro caso,} \end{cases} \tag{2.17}$$

donde las condiciones $cond_1$ y $cond_2$ son una función del estado de los vecinos y donde usualmente $\rho_{act_2} = \rho_{dec_2} = 0$, si bien esta condición puede relajarse para introducir ruido en el modelo.

Puesto que todos los métodos presentados parten de la existencia de una probabilidad de transición para cada nodo, es necesario poder transformar estos ritmos de vuelta en probabilidades. Esta equivalencia se recoge en la ecuación

$$w_i\left(y_i^{(t)} \mid x_{i,\partial i}^{(t)}\right) = \rho_{x_i \longrightarrow \tilde{y_i}} \left(1 - \delta_{x_i,y_i}\right) \tau + \left(1 - \rho_{x_i \longrightarrow \tilde{y_i}} \tau\right) \delta_{x_i,y_i}, \tag{2.18}$$

donde w_i es la probabilidad de transición entre estados de la ecuación 2.11, $\rho_{x_i \longrightarrow \tilde{y_i}}$ es el ritmo de transición de x_i a $y_i \neq x_i$, δ_{x_i,y_i} es la delta de Kroenecker y τ es un parámetro que regula la ventana temporal con la que se discretiza la evolución continua basada en ritmos (que se corresponde con el límite $\tau \longrightarrow 0$). Notar que las probabilidades de transición así contruidas están normalizadas sobre cada uno de los nodos independientemente del valor de τ, puesto que

$$\sum_{y_i} w_i\left(y_i^{(t)} \mid x_{i,\partial i}^{(t)}\right) = 1 - \rho_{x_i \longrightarrow \tilde{y_i}} \tau + \rho_{x_i \longrightarrow \tilde{y_i}} \tau = 1. \tag{2.19}$$

Más aún, cabe destacar que τ escala los ritmos para las transiciones, con lo que su valor determina el balance entre la probabilidad de que el nodo cambie de estado, o permanezca en su estado actual. Dado que se está tratando con probabilidades, las w_i deben ser no negativas, lo que observando la ecuación 2.18 introduce la condición $\rho_{x_i \longrightarrow \tilde{y_i}} \tau < 1$. El ρ más alto con el que se va a trabajar es 10, con lo que se va a fijar $\tau = 0.01$ quedando siempre dentro de dicha condición.

2.4.1. Modelo de tres nodos (modelo de juguete)

El primer modelo que se va a estudiar consta tan solo de tres nodos: A, B, C. El nodo A es activado por C e inhibido por B; el nodo B es activado por A y por C, y el nodo C es activado por A o por B. El grafo asociado se puede encontrar en la figura 2. Las relaciones entre los nodos vienen dadas por

$$\begin{aligned} A\colon & \begin{cases} \rho_{act} &= \rho_{u_1} \text{ si } (C \wedge B^c),\ 0 \text{ en otro caso} \\ \rho_{dec} &= \rho_{d_1} \text{ si } B,\ 0 \text{ en otro caso} \end{cases} \\ B\colon & \begin{cases} \rho_{act} &= \rho_{u_2} \text{ si } A,\ 0 \text{ en otro caso} \\ \rho_{dec} &= \rho_{d_2} \text{ si } A^c,\ 0 \text{ en otro caso} \end{cases} \\ C\colon & \begin{cases} \rho_{act} &= 0 \\ \rho_{dec} &= \rho_{escape} \text{ si } (A^c \wedge B^c),\ 0 \text{ en otro caso,} \end{cases} \end{aligned} \tag{2.20}$$

donde ρ_{u_i} y ρ_{d_i} son los parámetros correspondientes a los ritmos de activación y de decaimiento (respectivamente) de los correspondientes nodos y X^c es la aplicación del operador lógico "NOT"($\neg$) sobre el nodo X.

Dado que en este caso solo se tienen ocho posibles estados, resulta sencillo seguir las trayectorias (ver figura 2), con lo que se puede comprobar que existe un único punto fijo ([000]). Esto hace que la entropía de equilibrio sea 0 para cualquier condición inicial.

No obstante, cabe destacar que también existe un ciclo que decae al punto fijo cuando se pasa del estado [001] al [000]. La probabilidad de esta transición viene dada por ρ_{escape}, con lo que modificando los valores de este parámetro se puede modificar el tiempo de permanencia esperado en el ciclo hasta hacerlo infinito (en el límite $\rho_{escape} \longrightarrow 0$). En este caso límite, el sistema evolucionaría entre los cuatro estados del ciclo, con lo que el valor de los nodos no estaría fijado, lo que daría lugar a una entropía de equilibrio no nula.

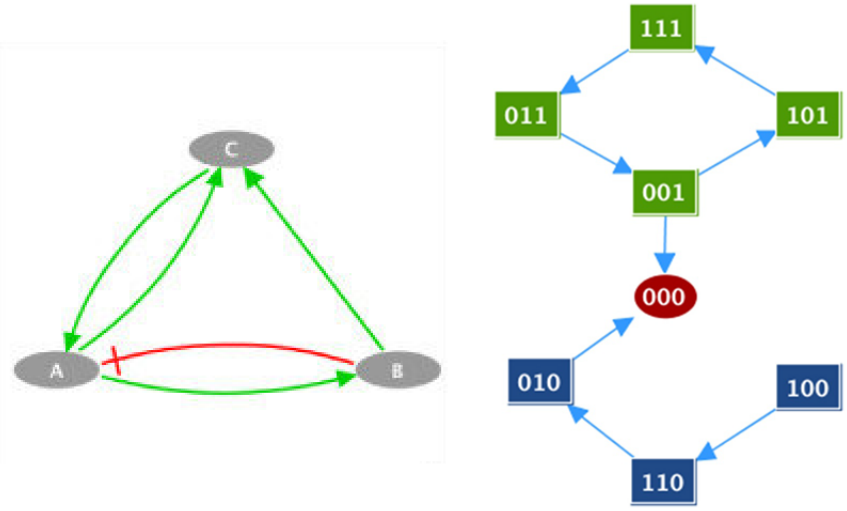

Figura 2: A la izquierda, grafo asociado al sistema del modelo de juguete. A la derecha, grafo de transiciones para los posibles estados. Notar que tan solo se muestran las transiciones posibles, si bien al haber introducido aleatoriedad en el modelo en algunos casos existe una probabilidad no nula de mantenerse en el estado actual. El orden de los nodos en los estados del diagrama sería $[ABC]$. Imágenes obtenidas de [5].

2.4.2. Modelo del ciclo celular

Este modelo está compuesto de diez nodos y describe los mecanismos que controlan la actividad de diferentes complejos CDK/ciclinas, quienes se encargan de la regulación de la dinámica en el ciclo celular. Las relaciones concretas entre los nodos se recogen en el apéndice D, así como su grafo asociado. En este caso, el grafo de la red de transiciones, análogo al que aparece a la derecha en la figura 2 para el modelo de tres nodos, tiene 1024 nodos. Así, resulta ya muy complicado obtener información de él, con lo que cobran importancia los métodos de estudio presentados anteriormente.

3. Resultados

Como ya se ha comentado, en este trabajo se van a analizar computacionalmente distintas redes booleanas mediante tres métodos diferentes con el fin de obtener información relevante sobre las mismas, a la vez que se examina el comportamiento del CVM, desde el punto de vista de la precisión de los resultados. En concreto, se quieren estudiar las virtudes y limitaciones del CVM, empleando como referencia para comparar los métodos de Monte Carlo y de la matriz de transferencia. Cabe mencionar que, mientras en los trabajos previos se hacian comparaciones con los resultados de métodos de Monte Carlo cinéticos en tiempo continuo obtenidos con el programa *MaBoSS* [5] (un código bien establecido y de referencia), aquí utilizamos un método de Monte Carlo en tiempo discreto, que se ha construido para reproducir de forma precisa las probabilidades de salto utilizadas en el CVM y en la evolución exacta.

El código desarrollado para estos fines se puede consultar en el repositorio de GitHub presentado en el apéndice E, y está compuesto por dos módulos acoplados. Por un lado, está el código desarrollado inicialmente en [3] corregido frente a la aparición de comportamientos caóticos, ampliado para incluir más métodos aparte del método de variación de clúster y generalizado para poder recibir cualquier tipo de red. Por otro lado, se tiene un módulo encargado de procesar cualquier red, generar los archivos necesarios para las simulaciones y procesar los resultados posteriores.

3.1. Modelo de tres nodos (modelo de juguete)

En primer lugar, se analizaron los resultados arrojados por el CVM aplicados sobre el modelo de tres nodos descrito en el apartado 2.4.1. Observando el grafo de transiciones de la figura 2 se puede ver que los dos tipos de trayectorias relevantes son la que parte de $[ABC] = [100]$ para llegar al atractor, y la que parte de cualquiera de las condiciones iniciales que componen el ciclo transitorio (por ejemplo $[ABC] = [111]$). Asimismo, como se ha comentado al presentar las ecuaciones que gobiernan la evolución de los nodos, el parámetro más influyente de este modelo es ρ_{escape}, ya que determina la probabilidad de abandonar el ciclo.

Usando las condiciones iniciales mencionadas y fijando $\rho_{escape} = 10$, se obtuvieron los resultados de la figura 3. En ella se puede ver cómo en el caso de comenzar en [100] los valores de la entropía exacta, del método de Monte Carlo (promediando 10000 trayectorias) y del CVM coinciden; mientras que partiendo de [111], los resultados de la simulación de Monte Carlo y de la solución exacta son iguales, pero difieren ligeramente de los del CVM. La razón de que para la condición inicial [100] el CVM sea un método exacto reside en el hecho de que en toda la trayectoria hasta llegar al atractor [000], el valor de C nunca es 1, con lo que el modelo es efectivamente un modelo de dos nodos. En ese caso, y dado que A y B son vecinos entre sí, se tiene que el clúster Q (asociado a cualquiera de los nodos) coincide con el sistema completo, con lo que realmente no existe un truncamiento de la serie de cumulantes de la entropía, y el valor obtenido por el CVM es exacto.

Cabe mencionar que, en este caso, se han usado para comparar con los resultados del CVM tanto el método de Monte Carlo como la solución exacta, obteniendo que los resultados de ambos son consistentes. Como se ha comentado anteriormente, en este trabajo se va a trabajar con redes de tamaño pequeño, con lo que el método basado en la matriz de transición tiene un coste computacional asequible y será el empleado para obtener los valores de referencia. No obstante, es importante disponer de la posibilidad de hacer simulaciones de Monte Carlo, puesto

que nos permitirán, por un lado, poder analizar trayectorias concretas sobre las redes, y por otro, extender los análisis aquí realizados a sistemas con un número de nodos mucho mayor.

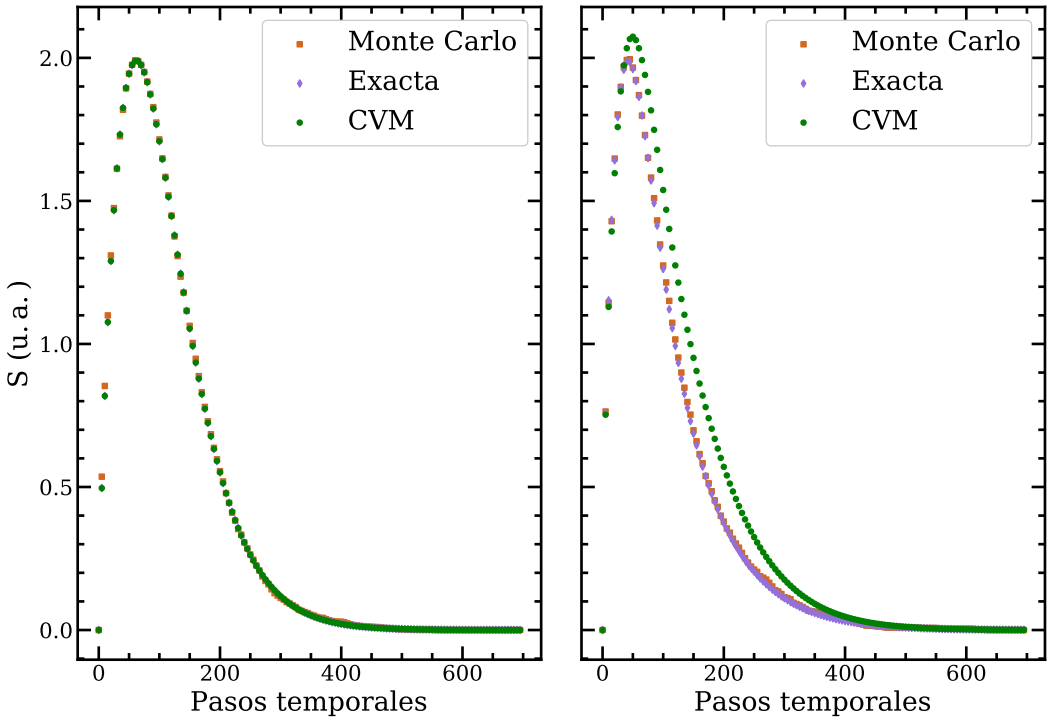

Figura 3: Comparación de los resultados de la simulación del modelo de juguete para dos condiciones iniciales fijando $\rho_{escape} = 10$. A la izquierda, los resultados obtenidos para la condición inicial $[ABC] = [100]$, y a la derecha, los obtenidos para la condición incial [111].

Dado que la condición inicial [111] está contenida en el ciclo transitorio, esta se usó para estudiar el efecto del valor de ρ_{escape} en la evolución de la entropía. De esta manera, se obtuvo la figura 4. En ella se puede ver cómo al reducir el valor de ρ_{escape} la entropía parece estabilizarse en un valor no nulo, lo que indica la existencia de un ciclo estable. No obstante, este ciclo es tan solo transitorio, puesto que, como se observa en la gráfica, pasados suficientes pasos temporales, la entropía comienza a descender, llegando eventualmente a 0 para tiempos muy grandes.

Aunque en este caso el ciclo se puede obtener analíticamente, conviene desarrollar las herramientas que se usarán cuando las redes sean demasiado grandes para poder determinar trayectorias concretas fácilmente. Así, con el fin de estudiar más a fondo este ciclo, se usó el método de Monte Carlo para simular una trayectoria partiendo de [111] (para $\rho_{escape} = 0.05$) durante 10000 pasos temporales. De esta forma, se obtuvo la figura 5 en la que se muestran los primeros 500 pasos de la trayectoria y el esquema de transiciones entre los estados del ciclo. En dicho esquema, el tamaño de los nodos (que aquí representan estados completos) es proporcional al número de pasos permanecido en ellos, y la anchura de los ejes es proporcional a la probabilidad de la transición asociada. Analizando la trayectoria se puede ver cómo en la mayoría de los pasos temporales el sistema no realiza ninguna transición, si bien el tiempo de estancia en cada estado varía. Esto se puede observar también en el hecho de que no todos los nodos del grado de transiciones tienen el mismo tamaño.

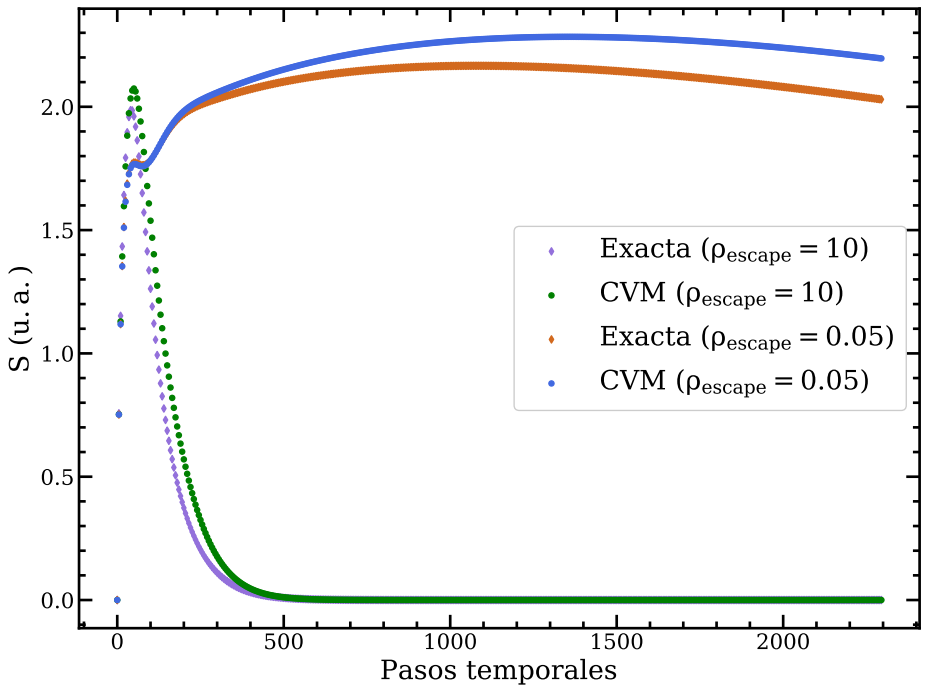

Figura 4: Comparación de los resultados de la entropía para dos valores de ρ_{escape} partiendo de la condición inicial $[ABC] = [111]$. En morado/verde, los resultados correspondientes a $\rho_{escape} = 10$, y en azul/naranja, los resultados para $\rho_{escape} = 0.05$.

3.2. Modelo de ciclo celular

Una vez comprobada la correcta implementación de los métodos sobre un modelo sencillo, se pasó a analizar un modelo más complejo compuesto de diez nodos (véase apartado 2.4.2 y apéndice D). Notar que a lo largo de la sección se trabajará con los estados del sistema escritos como números binarios y como números enteros. La forma de pasar de una representación a otra, y de traducir ambas en una configuración del sistema, se encuentra en el apéndice D.1.

Dado que en este caso no se disponía de una visualización sencilla del grafo de transición, no se conocían las condiciones iniciales que dan lugar a los distintos tipos de trayectorias. Por ello, se realizó en primer lugar una exploración de todas las condiciones iniciales, de forma que se pudieran conocer cuáles daban lugar a resultados del CVM más similares a la evolución exacta, así como cuáles son los tipos posibles de atractores de la dinámica.

3.2.1. Inestabilidad computacional y comportamiento caótico

Tras una primera exploración, se encontraron varias condiciones iniciales en las que el valor asintótico de la entropía del CVM distaba considerablemente del valor de la entropía exacta.

Para estudiar la causa de estas discrepancias, se graficaron las trayectorias completas de algunas de estas condiciones iniciales, observándose principalmente dos perfiles de discrepancias (ver figura 6). Por un lado, en algunas trayectorias aparecía repentinamente un pico de entropía para la simulación del CVM que no se observaba en la evolución exacta, y por otro lado, en otras trayectorias se observaba una entropía oscilante en el CVM, mientras que la entropía exacta tendía progresivamente a 0. Igualmente, se observó que para determinadas trayectorias del CVM había puntos con entropía negativa, lo que parecía implicar la existencia de errores en la simulación. Con el fin de poder comparar, el mismo código se compiló con otro compilador y

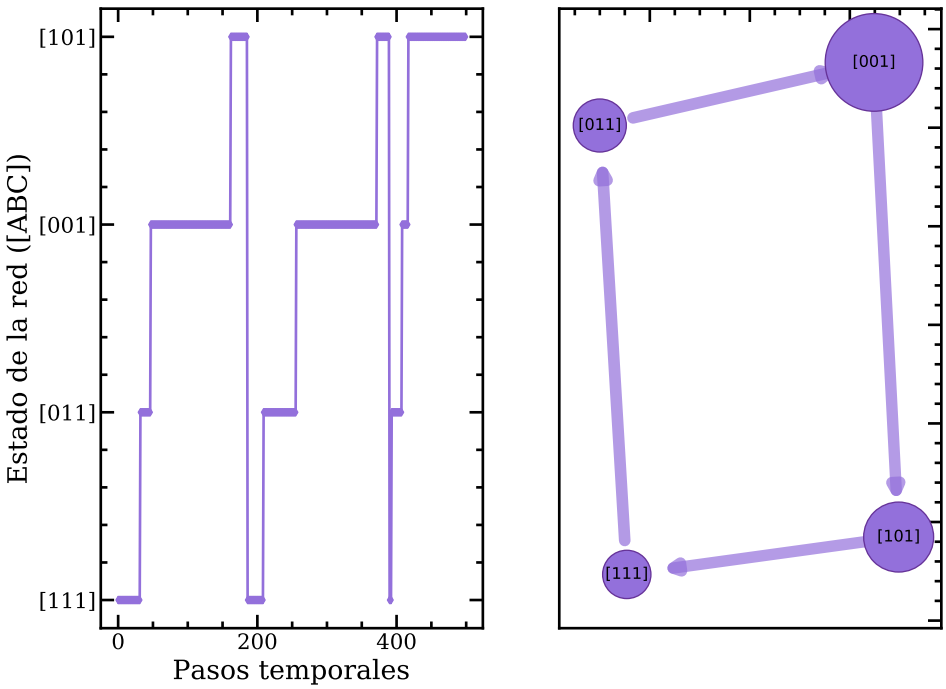

Figura 5: A la izquierda, primeros pasos temporales de una trayectoria partiendo de $[ABC] = [111]$ obtenida mediante el método de Monte Carlo para $\rho_{escape} = 0.05$. A la derecha, grafo de transiciones entre los estados del ciclo para esa misma simulación considerando 10000 pasos temporales. Los nodos de este grafo representan estados del sistema y su tamaño es proporcional al tiempo de estancia de cada estado. Por su parte, los ejes tienen una anchura proporcional a la transición que representan.

se ejecutó en un sistema operativo diferente, observándose que en este caso no se encontraban las discrepancias mencionadas.

Es común al trabajar con C encontrarse con pequeñas discrepancias debidas al compilador, puesto que es un lenguaje en el que ciertas operaciones (por ejemplo los redondeos a la máxima precisión permitida) dependen de la implementación que se haga en cada compilador. En esta línea, se han desarrollado trabajos para conseguir compiladores que realicen las operaciones en coma flotante de manera estandarizada como se recoge en [16]. No obstante, las diferencias tan significativas encontradas aquí mostraban, por un lado, la fuerte sensibilidad del método frente a perturbaciones, y por otro, la necesidad de implementar un algoritmo más robusto.

La sensibilidad del método se puede apreciar fácilmente en la fila inferior de la figura 6, donde una ligera discrepancia en torno al paso temporal 100 da lugar a evoluciones totalmente opuestas, lo que sugiere un comportamiento caótico del algoritmo empleado.

Para conseguir que el método fuera consistente, independientemente del proceso de compilación empleado, surgió la necesidad de desarrollar un programa más robusto frente a la aparición de estas perturbaciones, para lo que se siguieron dos estrategias. En primer lugar, se forzó la normalización de todos los clústeres para cada paso temporal, de forma que ninguna de las probabilidades marginales se desviara en gran medida. Esta modificación estabilizaba ciertas trayectorias, si bien otras seguían presentando comportamientos erráticos. Así, se introdujo como segunda modificación el promediado del valor de los clústeres sobre todas las posibles marginalizaciones. En concreto, para el clúster V se promedió el resultado obtenido en la ecuación C.5

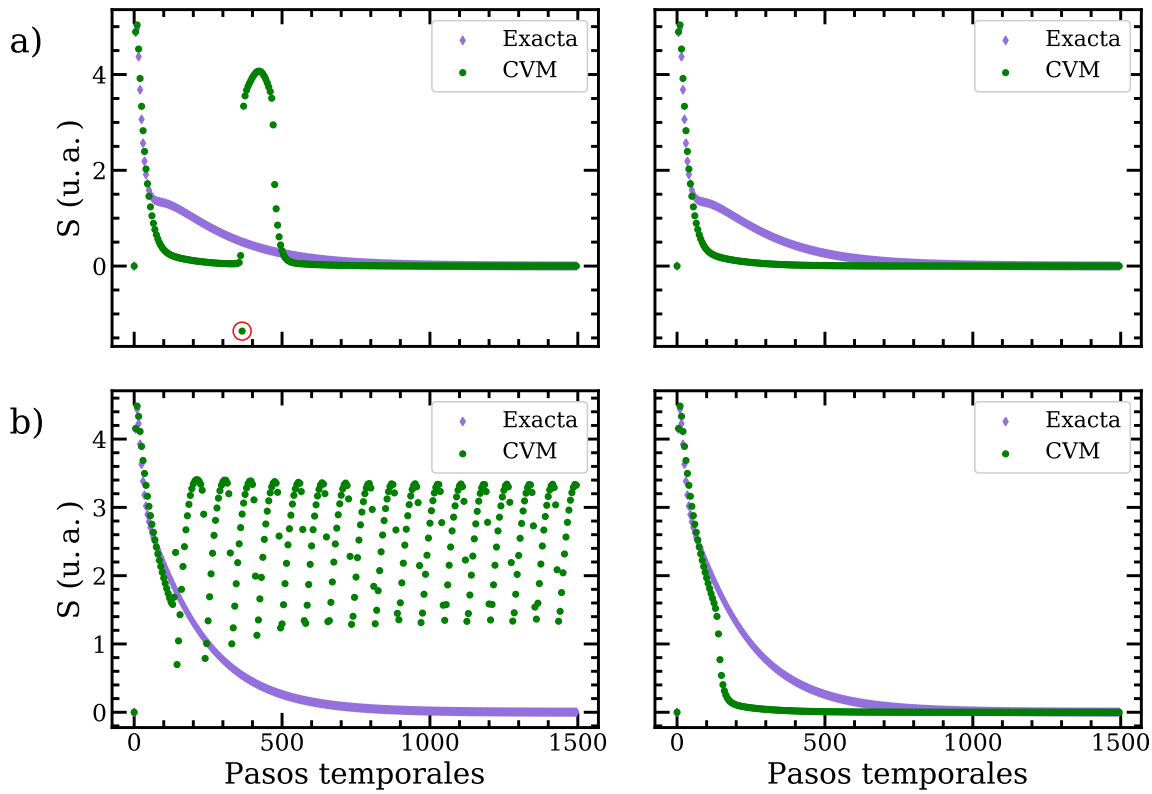

Figura 6: Comparación de los resultados de la simulación para dos condiciones iniciales, empleando dos compiladores y sistemas operativos diferentes. En la parte superior, los resultados asociados a las condición inicial [0110001101], y en la parte inferior los correspondientes a [0011100011]. A la izquierda, los resultados obtenidos con la versión 6.3.0 de *gcc* para un modelo de hilos *win32* (sistema opetativo Windows); y a la derecha, los resultados obtenidos con la versión de *gcc* 14.2.0 en un modelo de hilos *posix* (sistema operativo Linux, distribución Ubuntu). En rojo, en la gráfica superior izquierda, se marca el punto de entropía negativa encontrado.

con los valores de marginalizar T

$$\sum_{x_j} T^{(t)}_{i,ij}\left(y_i, x_{i,j}\right) = V^{(t)}_i\left(y_i, x_i\right) \quad \forall i, \forall j \in \partial_i \tag{3.1}$$

y U

$$\sum_{y_j} U^{(t)}_{ij,i}\left(y_{i,j}, x_i\right) = V^{(t)}_i\left(y_i, x_i\right) \quad \forall i, \forall j \in \partial_i; \tag{3.2}$$

y para el clúster Z, se promedió el resultado de la ecuación C.8 con el resultado de la marginalización de U

$$\sum_{x_i} U^{(t)}_{ij,i}\left(y_{i,j}, x_i\right) = Z^{(t+1)}_{ij}\left(y_{i,j}\right) \quad \forall i, \forall j \in \partial_i. \tag{3.3}$$

De esta manera se consiguió que todas las trayectorias estuvieran estabilizadas, con lo que se pudo pasar a la exploración de las condiciones iniciales.

3.2.2. Exploración de las condiciones iniciales

Una vez comprobada la robustez del código frente a errores de redondeo, se pasó a realizar un análisis del comportamiento asintótico de la red sobre todas las condiciones inciales. Para ello, se emplearon tanto el CVM como el método basado en la matriz de transferencia, de forma que se pudiera comparar la entropía aproximada con la exacta. Se observó que, para la mitad (512) de las posibles condiciones iniciales, la entropía asintótica era 0, tanto en el caso exacto como en el aproximado, mientras que para la otra mitad se tenía que $S_{CVM} \longrightarrow 3.277$ y $S_{Exacta} \longrightarrow 3.464$. Este hecho sugiere la existencia de, al menos, un estado atractor (lo que daría lugar a la entropía nula) y un ciclo estable (en el que la entropía sería constante, pero no nula). Estos dos mismos tipos de comportamientos estacionarios se encuentran en las conclusiones de [5], si bien en ese caso la proporción de condiciones iniciales que van a parar a cada uno difiere ligeramente (el 48 % van a parar al punto fijo, y el 52 % al ciclo atractor, frente al 50 %-50 % obtenido aquí).

A la hora de extraer conclusiones de manera cualitativa, la discrepancia del 5.4 % entre el valor exacto y el del CVM es aceptable, si bien podría quedarse corta si se quisieran extraer conclusiones cuantitativas con mayor precisión.

A fin de obtener más información sobre el comportamiento transitorio, se calculó la integral sobre los pasos temporales de la diferencia cuadrática entre la entropía exacta y la entropía del CVM para todas las condiciones iniciales. Si bien el CVM es un método inicialmente ideado para obtener resultados asintóticos, con esta medida se puede comprobar si de las estimaciones del CVM a lo largo de la simulación también se pueden extraer conclusiones aproximadas. Los resultados obtenidos se separaron en función del valor asintótico de la entropía obteniendo los resultados de la figura 7.

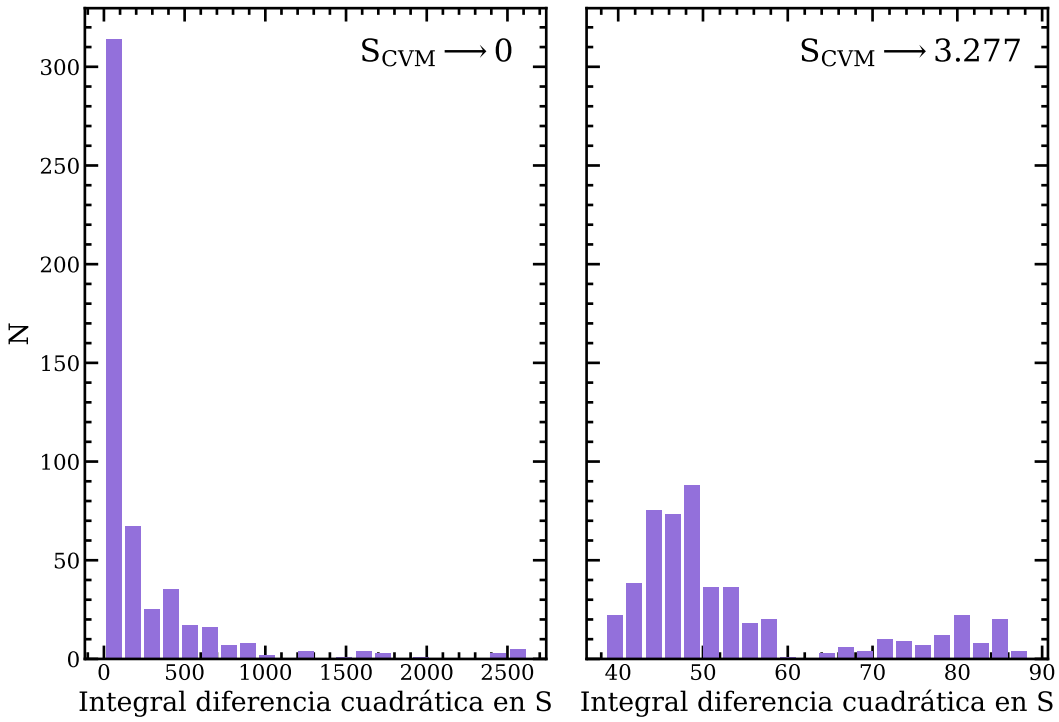

Figura 7: Distribución de los valores de la integral sobre los pasos temporales de la discrepancia cuadrática entre la entropía exacta y la entropía del CVM. A la izquierda, la distribución solo para aquellas condiciones iniciales en las que la entropía asintótica es 0. A la derecha, la distribución cuando la entropía asintótica es no nula.

De esta figura se observa, por un lado, que en los casos en los que la entropía asintótica es nula la gran mayoría de condiciones iniciales sigue una trayectoria similar a la que se obtiene mediante la matriz de transferencia. Sin embargo, existe una cierta proporción de condiciones iniciales para las que las discrepancias van creciendo hasta alcanzar valores 25 veces superiores a las máximas diferencias alcanzadas cuando la entropía asintótica no se anula. Para analizar la causa del creciente valor de las discrepancias se graficó la evolución de cuatro condiciones iniciales con valores de discrepancias crecientes, obteniendo la figura 8. En ella se puede ver cómo la entropía del CVM sigue siempre una forma similar, presentando un pico inicial en los primeros pasos temporales para luego decaer rápidamente a 0. Por su parte, la entropía exacta puede presentar una mayor variedad de patrones, incluyendo dobles picos, picos más tardíos, y sobre todo decae más lentamente a 0.

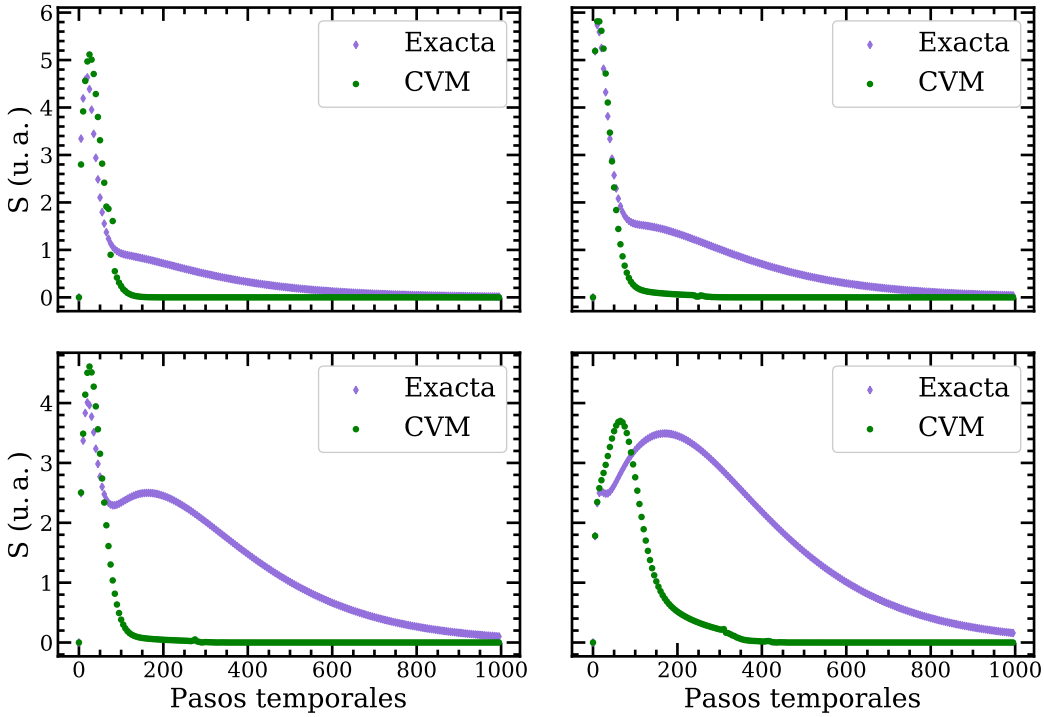

Figura 8: Comparación de los resultados de la simulación del modelo de ciclo celular para cuatro condiciones iniciales cuya entropía estacionaria tiende a cero. De izquierda a derecha, los resultados obtenidos para las condiciones iniciales [0111000111], [0111110111], [0101010111] y [0100010010].

Por otro lado, cuando la entropía asintótica es no nula, se observa que la dispersión de las discrepancias es mucho menor ya que todas se encuentran entre 40 y 90. No obstante, las discrepancias quedan lejos de 0, con lo que hay diferencias relevantes entre las evoluciones de las entropías. En concreto, dado que se observan dos picos (uno en torno a 50 y otro en torno a 80) en los que se concentran los valores, se espera encontrar dos tipos de perfiles para la estabilización de la entropía. Esto se observa en la figura 9, donde se han elegido las condiciones iniciales de menor ([1010011101]) y mayor ([1101011011]) discrepancia. En dicha figura, se puede ver que, cuando la entropía crece mucho en los primeros pasos temporales, la entropía del CVM atraviesa un valle y se despega del valor de la entropía exacta, que presenta un pico, para después estabilizarse ambas. En cambio, cuando la entropía se mantienen acotada en los pasos iniciales, ambas entropías se mantienen más próximas y presentan un pico inicial, si bien la entropía del

CVM decae más rápidamente hasta alcanzar su valor de equilibrio.

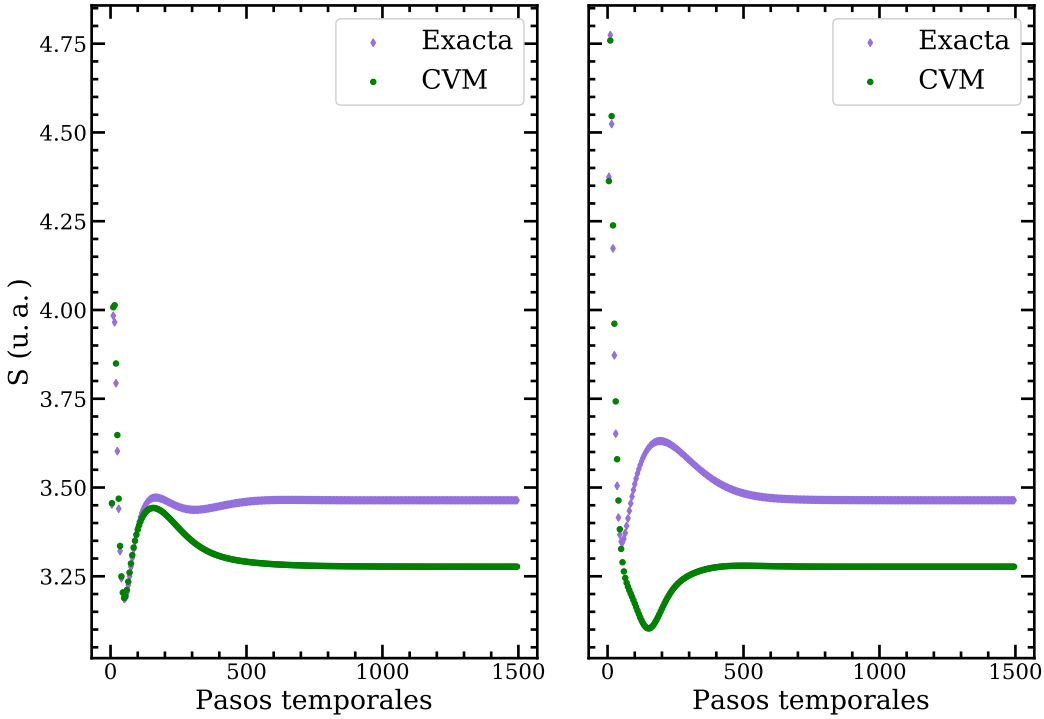

Figura 9: Comparación de los resultados de la simulación del modelo de ciclo celular para dos condiciones iniciales, cuya entropía estacionaria es no nula. A la izquierda, los resultados obtenidos para la condición inicial [1010011101], y a la derecha, los obtenidos para la condición inicial [1101011011].

Como se ha comentado anteriormente, el hecho de que la entropía tienda a 0 en determinadas trayectorias parece corresponder con la existencia de estados atractores, y el hecho de que su valor de equilibrio sea en ocasiones no nulo, con la existencia de ciclos estables. Cabe destacar que también es posible la existencias de ciclos transitorios como parece deducirse de la evolución de la entropía exacta en la gráfica inferior izquierda de la figura 8, si bien estos son más difíciles de detectar.

Para estudiar más a fondo los atractores, se analizaron las probabilidades obtenidas mediante de los distintos nodos de estar activados en el equilibrio. Con ello se observó que el estado final de todas las trayectorias en las que la entropía se anulaba era el mismo para ambos métodos ([0000101001]), y se correspondía con el mismo estado atractor descrito en [5]. Por su parte, en las trayectorias con entropía de equilibrio no nula se vió que, para un mismo método, las probabilidades de equilibrio de todos los nodos también eran iguales entre trayectorias con diferentes condiciones iniciales, si bien los valores concretos diferían entre métodos. Con el fin de conocer mejor las trayectorias seguidas, se realizó una simulación de Monte Carlo de 15000 pasos temporales partiendo de una de las condiciones iniciales ([1101001000]) que daba lugar a entropía no nula.

Mediante este análisis se vió que no existía un ciclo estable como se había pensado, sino que realmente el estado de equilibrio consistía en un conjunto de ciclos transitorios acoplados. El grafo de las transiciones seguidas a lo largo de la trayectoria (figura 10), permite observar la existencia de ciertos estados en los que el tiempo de permanencia es mayor que en el resto: 269, 261, 513, 547, 545, 5, 13, 769, 897. Dichos estados, traducidos a sus correspondientes configuraciones de

nodos, se corresponden con conjuntos de nodos activos descritos en [5] como estados cuasiestables de la distribución estacionaria de las trayectorias que no van al punto fijo. La única excepción es el estado 13 ([1011000000]) que no se recoge en la enumeración del artículo. Notar, no obstante, que allí se incluyen solo los estados mayoritarios resultantes de un gran número de simulaciones, mientras que aquí solo se ha considerado una trayectoria, con lo que es posible que al promediar más trayectorias, el peso de este estado se diluya.

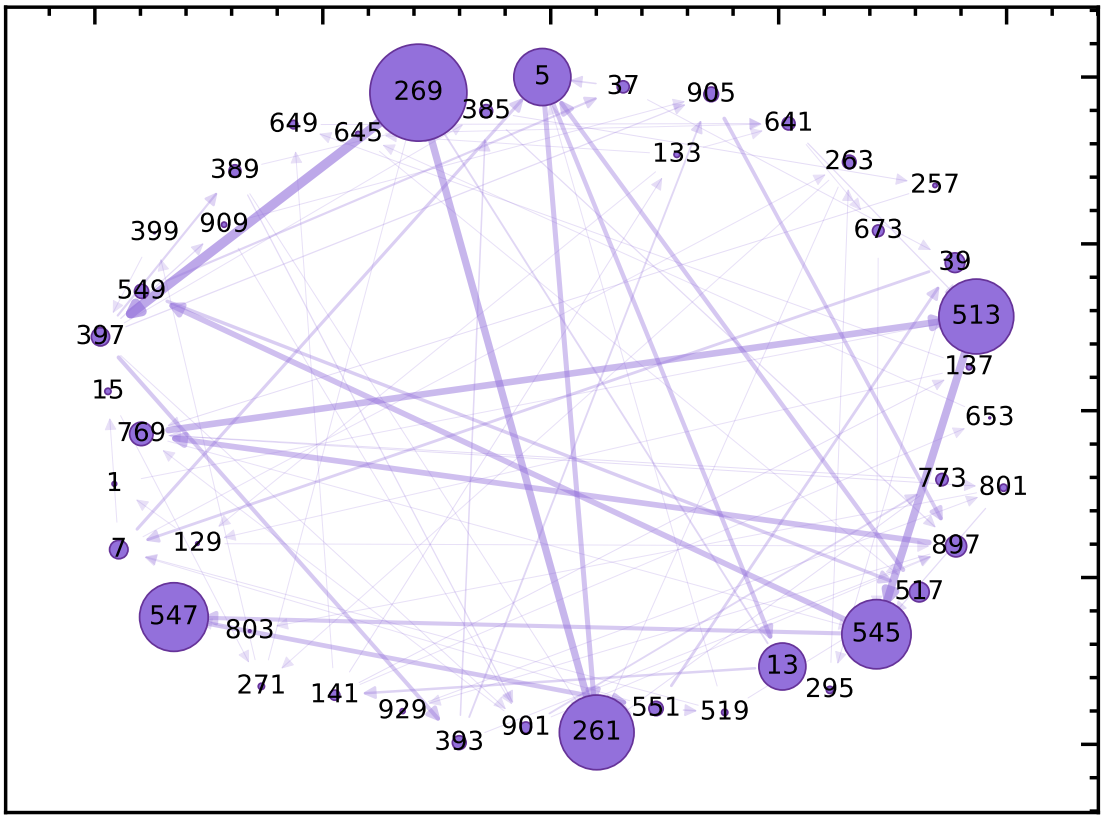

Figura 10: Grafo de transiciones entre los estados visitados por la trayectoria de Monte Carlo partiendo de la condición inicial [1101001000], considerando 15000 pasos temporales. Los nodos de este grafo representan estados del sistema y su tamaño es proporcional al tiempo de estancia de cada estado. Por su parte, los ejes tienen una anchura proporcional a la transición que representan.

4. Conclusiones y trabajo futuro

En este trabajo, se han presentado en primer lugar los conceptos de redes de regulación genética y de redes booleanas, explicando sus diferentes representaciones y los distintos tipos de evoluciones que se pueden definir sobre ellas, y haciendo especial énfasis en la evolución probabilística.

Tras esto, se ha presentado el método CVM, tanto en su versión clásica (inicialmente propuesta por Kikuchi [12], y adaptada por Heskes y colaboradores [13]), como en la versión dinámica planteada por Pelizzola y Pretti [2]. Asimismo, se han introducido los métodos (Monte Carlo y evolución mediante la matriz de transición) contra los que se han comparado los resultados obtenidos por el CVM. A continuación, se han presentado los modelos de redes de regulación genética sobre los que se han aplicado los métodos anteriores, poniendo especial atención en la

forma de transformar el enfoque de tiempo continuo y ritmos de transición de [5] (de dónde se han extraído los modelos) en una evolución discreta basada en probabilidades de transición.

Este método y estos modelos ya se habían considerado en una investigación previa [3, 4], si bien en ese caso el análisis se centró en la evolución temporal a partir de ciertas condiciones iniciales fijas, y la referencia para comparar fue el programa *MaBoSS* [5]. Aquí, tras llevar a cabo un extenso proceso de reescritura del código previo para poder emplearlo en redes arbitrarias, facilitando el estudio de perturbaciones y modificaciones estructurales, se ha profundizado en el estudio del comportamiento del algoritmo. En particular, se ha detectado una inestabilidad, por pérdida de precisión numérica, del algoritmo de Pelizzola y Pretti, en su implementación de acuerdo al esquema publicado en [2] (véase apéndice C), la cual se ha solucionado forzando las relaciones de marginalización entre clústeres. Hay que destacar que el algoritmo está originalmente ideado para grafos no dirigidos, así que esta inestabilidad podría estar relacionada con el uso actual en grafos dirigidos.

Con el modelo de juguete, se ha observado un caso en el que el CVM dinámico es exacto, lo cuál se da cuando alguno de los clústeres maximales coincide con el sistema total, y se han analizado los distintos tipos de estados estacionarios que puede tener una red, observando cómo se traduce cada uno en un comportamiento de la entropía estacionaria.

En cuanto al modelo de ciclo celular, se han observado dos atractores de la dinámica, ambos con una cuenca de atracción compuesta por la mitad de las condiciones iniciales. Por un lado, se ha encontrado un punto fijo, que se puede reconocer mediante la evolución de la entropía, por ir esta a 0. Dicho punto fijo se ha comprobado que corresponde con el mencionado en [5].

Por otro lado, se ha encontrado la existencia de un conjunto amplio de estados, del cuál una trayectoria no escapa una vez ha entrado, que son recorridos mediante ciclos transitorios acoplados. Este estado estacionario da lugar a una entropía no nula, puesto que el sistema no se queda estático en ningún estado. De este conjunto, se han obtenido los estados con mayores tiempo de permanencia y se ha observado una buena concordancia con los resultados de [5].

En definitiva, podemos concluir que el CVM reproduce correctamente, aunque no de forma cuantitativa, los resultados exactos, siendo fiable para tiempos largos, mientras que su exactitud en tiempos cortos tiene una fuerte dependencia de las condiciones iniciales.

Este trabajo abre otras posibles vías de investigación. El hecho de desarrollar un código apto para procesar redes arbitrarias permite analizar rápidamente un gran número de sistemas, introducir modificaciones sobre ellos, y extraer conclusiones. Más aún, el esfuerzo hecho para estabilizar el código permite minimizar el riesgo de comportamientos caóticos inesperados en sistemas sobre los que a priori esto podría ser difícil de detectar.

En concreto, un sistema sobre el que sería interesante poder obtener resultados sería una red con estructura de árbol. Este sistema es el más sencillo (no trivial) que se puede construir, y el CVM clásico debería dar resultados exactos al aplicarse sobre él. No obstante, cuando se trabaja con el CVM dinámico, la dimensión temporal juega un papel crucial, por lo que la estructura de árbol en ese caso se pierde y nada asegura que los resultados del CVM deban ser exactos.

Por último, la combinación del método CVM con la posibilidad de procesar redes arbitrarias, abre la puerta al análisis de sistemas de gran dimensionalidad, ya que poporciona un método rápido y de una precisión aceptable con el que poder extraer conclusiones sobre los estados estacionarios de las mismas. Un ejemplo de estos sistemas sería el descrito en [17], mediante el cual se pueden obtener sistemas arbitrariamente grandes repitiendo una unidad mínima de nodos e interacciones.

Referencias

[1] J. Yedidia, W. Freeman e Y. Weiss, «Constructing free-energy approximations and generalized belief propagation algorithms», IEEE Transactions on Information Theory **51**, 2282 (2005).

[2] A. Pelizzola y M. Pretti, «Variational approximations for stochastic dynamics on graphs», Journal of Statistical Mechanics: Theory and Experiment **7**, 073406 (2017).

[3] P. Pérez Lázaro, P. Bruscolini y J. Sanz Remón, «Estudios de redes de regulación genéticas con modelos discretos», (2021).

[4] P. Pérez Lázaro y P. Bruscolini, «Desarrollo de una aproximación variacional para la cinética de procesos markovianos en grafos. Aplicación a redes de regulación genética.», (2022).

[5] G. Stoll, E. Viara, E. Barillot y L. Calzone, «Continuous time Boolean modeling for biological signaling: application of Gillespie algorithm», BMC systems biology **6**, 1 (2012).

[6] S. Z. S. Mohammad, «Biological Networks: An Introductory Review», Journal Of Proteomics And Genomics Research **2**, 41 (2018).

[7] B. A. Hall y A. Niarakis, «Data integration in logic-based models of biological mechanisms», Current Opinion in Systems Biology **28**, 100386 (2021).

[8] J. D. Schwab, S. D. Kühlwein, N. Ikonomi, M. Kühl y H. A. Kestler, «Concepts in Boolean network modeling: What do they all mean?», Computational and Structural Biotechnology Journal **18**, 571 (2020).

[9] J. Yedidia, W. Freeman, Y. Weiss et al., «Understanding belief propagation and its generalizations», Exploring artificial intelligence in the new millennium **8**, 0018 (2003).

[10] A. Pelizzola, «Cluster variation method in statistical physics and probabilistic graphical models», Journal of Physics A: Mathematical and General **38**, R309 (2005).

[11] G. An, «A note on the cluster variation method», Journal of Statistical Physics **52**, 727 (1988).

[12] R. Kikuchi, «A Theory of Cooperative Phenomena», Physical Review Journals **81**, 988 (1951).

[13] T. Heskes, K. Albers y H. Kappen, «Approximate Inference and Constrained Optimization», Conference on Uncertainty in Artificial Intelligence (2002).

[14] H. C. Nguyen y J. Berg, «Bethe–Peierls approximation and the inverse Ising problem», Journal of Statistical Mechanics: Theory and Experiment **2012**, P03004 (2012).

[15] E. Stoian, «Fundamentals and Applications of the Monte Carlo Method», Journal of Canadian Petroleum Technology **4**, 120 (1965).

[16] S. Boldo, J.-H. Jourdan, X. Leroy y G. Melquiond, «A Formally-Verified C Compiler Supporting Floating-Point Arithmetic», IEEE 21st Symposium on Computer Arithmetic (2013).

[17] R. Albert y H. G. Othmer, «The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster», Journal of Theoretical Biology **223**, 1 (2003).

A. Cadenas de Markov

Los métodos empleados para el estudio de las redes booleanas de este trabajo se basan en las propiedades derivadas de la modelización del sistema como cadenas de Markov, con lo que es necesario profundizar sobre sus propiedades.

Como concepto previo, se debe hablar de los procesos estocásticos, que son familias de variables aleatorias $\{X(t) \mid t \in T\}$ definidas sobre un espacio de probabilidad cualquiera indexadas con un parámetro t que varía en un conjunto de índices T.

En función del espacio de estados (S) en el que toman valor las variables aleatorias, se habla de proceso continuo (variables aleatorias continuas) o discreto (variables aleatorias discretas). Estos últimos son los más usuales y son el objeto de estudio en este trabajo, por lo que de ahora en adelante solo se presentarán los conceptos para procesos discretos (para procesos continuos son, por lo general, análogos). En función del rango de valores que toma t se hablará de procesos de tiempo continuo o procesos de tiempo discreto.

El conjunto de estados concretos observados a lo largo del tiempo se denomina realización del proceso, y cada cambio de estado se denomina transición.

Las cadenas de Markov son un tipo particular de procesos estocásticos en los que el estado del sistema en el futuro solo depende del estado del sistema actualmente, con lo que es independiente de lo que haya ocurrido en el proceso en el pasado (propiedad de Markov).

Dentro de las cadenas de Markov, un tipo de especial interés son las cadenas homogéneas, que se caracterizan porque las propiedades de transición entre estados son independientes del tiempo, es decir, se puede definir una matriz de transición

$$P = (p_{ij})_{i,j \in T}$$

donde p_{ij} es la probabilidad de pasar del estado i al j.

Notar que las filas de la matriz de transición (también llamada matriz estocástica) suman 1, puesto que estarían expresando la probabilidad de que, dado un estado, en el siguiente instante el sistema estuviera en otro estado cualquiera (incluído él mismo).

A.1. Cadenas de Markov en tiempo continuo

Cuando el conjunto T sobre el que se indexa la familia de variables aleatorias es un intervalo continuo de $\mathbb{R}$, se tiene un proceso estocástico en tiempo continuo.

En este caso, la propiedad de Markov se expresa como

$$P(X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n, \ldots, X(t_1) = x_1) = P(X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n)$$

con $t_{n+1} > t_n > \ldots > t_1$ tales que $t_i \in \mathbb{R}$.

A.2. Cadenas de Markov en tiempo discreto

Se habla de cadena de Markov en tiempo discreto cuando el conjunto T que indexa la familia de variables aleatorias es un subconjunto de $\mathbb{N}$. En este caso, se suele usar la variable n que toma valores $0, 1, 2 \ldots$.

La propiedad de Markov se expresa como

$$P(X_{n+1} = x_{n+1} \mid X_n = x_n, X_{n-1} = x_{n-1} \ldots, X_0 = x_0) = P(X_{n+1} = x_{n+1} \mid X_n = x_n).$$

En este caso, resulta sencillo obtener una expresión exacta para la distribución de probabilidad a la que se llegaría partiendo de una distribución de probabilidad inicial sobre los estados.

Dado un vector $\mathbf{q}(0)$ de longitud el número de estados, que represente estas probabilidades iniciales, la distribución de probabilidad en un instante n viene dada por

$$\mathbf{q}(n) = \mathbf{q}(0) P^n. \tag{A.1}$$

B. Aproximación de la entropía para el CVM

Como se ha comentado en la sección 2.1, la aproximación que se realiza para la entropía en el CVM no es más que un truncamiento de la expansión de S como serie de cumulantes. Antes de pasar a la aproximación como tal, conviene definir ciertos conceptos que se usarán posteriormente. Notar que se ha seguido como guía el artículo de An [11] adaptando la demostración allí realizada.

En primer lugar, consideremos un conjunto $P = \{\alpha, \beta, \gamma, \dots\}$ sobre el que hay definida una relación binaria R reflexiva, transitiva y antisimétrica. Dicha relación se denomina orden parcial sobre P y se escribe $\leq$. El conjunto P se dice que está parcialmente ordenado (por ejemplo, $\mathbb{N}$ con la relación menor o igual).

Sea ahora la función $\zeta : P \times P \to \{0, 1\}$ tal que

$$\zeta(\beta, \alpha) = \begin{cases} 1 & \text{si } \beta \leq \alpha, \\ 0 & \text{en otro caso.} \end{cases} \tag{B.1}$$

Se define la función de Möbius μ del conjunto parcialmente ordenador P como la única función $\mu : P \times P \to \mathbb{Z}$ que satisface

$$\sum_{\alpha \leq \beta \leq \gamma} \zeta(\alpha, \beta)\, \mu(\beta, \gamma) = \delta(\alpha, \gamma), \tag{B.2}$$

donde $\delta(\alpha, \gamma)$ es la delta de Kronecker.

Consideremos por último, dos funciones $f, g : P \to \mathbb{R}$ cualesquiera tales que

$$f(\alpha) = \sum_{\beta \leq \alpha} g(\beta),$$

entonces se tiene que

$$g(\alpha) = \sum_{\beta \leq \alpha} f(\beta)\, \mu(\beta, \alpha), \tag{B.3}$$

donde $\alpha, \beta \in P$.

Una vez vistas las definiciones previas, se puede pasar ya a desarrollar la aproximación de la entropía. Consideremos un sistema formado por N nodos $L = \{X_1, X_2, \dots X_N\}$ y definamos el conjunto P como el conjunto de las partes de L, con lo que un clúster es un subconjunto cualquiera de L. Diremos que $\alpha \leq \beta$ si el $\alpha \subset \beta$. Esta relación de orden define un conjunto parcialmente ordenado en el que L es el clúster más grande.

Suponiendo una distribución de probabilidad sobre el sistema, se puede hablar de la distribución de probabilidad sobre los distintos clústeres obtenida como la marginalización de la distribución global. Así, tiene sentido hablar tanto de la entropía del sistema como de la entropía de cada clúster, calculándose en ambos casos de acuerdo a la ecuación 2.3. Notar que la entropía del clúster L (S_L) coincide con la entropía del sistema S,

A partir de la función de Möbius, se pueden definir las funciones

$$\tilde{S}_\alpha = \sum_{\beta \leq \alpha} \mu(\beta, \alpha)\, S_\beta, \tag{B.4}$$

que tienen como ventaja frente a sus respectivas S_α el hecho de que se espera que su valor vaya a 0 conforme el tamaño del clúster considerado sea mayor que la longitud de correlación.

Estas funciones $\tilde{S}_\alpha$ realmente no son más que los cumulantes, como se puede ver usando la inversión de Möbius (ecuación B.3) e identificando $f(\alpha) = S_\alpha$ y $g(\beta) = \tilde{S}_\beta$,

$$S_\alpha = \sum_{\beta \leq \alpha} \tilde{S}_\beta. \tag{B.5}$$

Así, recordando que $S = S_L$ se tiene que

$$S = S_L = \sum_{\alpha \leq L} \tilde{S}_\alpha = \sum_{\alpha \in P} \tilde{S}_\alpha \tag{B.6}$$

de manera exacta.

Se introduce ahora la aproximación usada en el CVM. De esta manera, se elige un conjunto de clústeres maximales $C = \{\gamma_1, \gamma_2, \ldots \gamma_k\}$ (es decir, ningún γ_i es subclúster de otro γ_j) y se mantienen solo los términos de la serie asociados con los subclústeres obtenidos mediante intersecciones de estos clústeres maximales. Denotando como P' el conjunto de los clústeres maximales y sus subclústeres, se tiene que

$$S \approx \sum_{\alpha \in P'} \tilde{S}_\alpha, \tag{B.7}$$

donde el truncamiento está justificado por la rápida convergencia de las $\tilde{S}_\alpha$ a 0 al aumentar el tamaño de α comentada anteriormente. Notar que esta aproximación es susceptible de fallar cuando la longitud de correlación del sistema sea comparable al tamaño de los clústeres maximales.

Sustituyendo la ecuación B.4 en B.7 y recordando la definición de la función ζ en B.1 se tiene que

$$S \approx \sum_{\alpha \in P'} \tilde{S}_\alpha = \sum_{\alpha \in P'} \sum_{\beta \leq \alpha} \mu(\beta, \alpha) S_\beta = \sum_{\alpha \in P'} \sum_{\beta \in P'} \mu(\beta, \alpha) \zeta(\beta, \alpha) S_\beta. \tag{B.8}$$

Finalmente, se puede intercambiar el orden de los sumatorios (son sumas finitas) para obtener

$$S \approx \sum_{\beta \in P'} c_\beta S_\beta, \tag{B.9}$$

donde se han definido los coeficientes c_β (que se suelen llamar números de Möbius) como

$$c_\beta = \sum_{\alpha \in P'} \mu(\beta, \alpha) \zeta(\beta, \alpha). \tag{B.10}$$

De esta manera se obtiene la aproximación de la entropía empleada en el CVM, así como un método de cálculo de los números de Möbius. No obstante, el cálculo de estos se puede simplificar empleando la definición de la función de Möbius μ dado que

$$\sum_{\beta \geq \alpha} c_\beta = \sum_{\gamma \in P'} \sum_{\alpha \leq \beta \leq \gamma} \zeta(\alpha, \beta) \mu(\beta, \gamma) = \sum_{\gamma \in P'} \delta(\alpha, \gamma) = 1 \Longrightarrow \sum_{\beta \geq \alpha} c_\beta = 1. \tag{B.11}$$

Así, resolviendo recursivamente sobre el tamaño de los clústeres se pueden obtener todos los coeficientes de manera sencilla.

C. Algoritmo de simulación del CVM cinético

Como se ha comentado, para el CVM cinético aplicado a sistemas provenientes de cadenas de Markov es común que exista un algoritmo sencillo que permita obtener el mínimo de la energía libre de Helmholtz. Dicho algoritmo, aplicado al caso de la aproximación PQR, se muestra a continuación, y se encuentra resumido en la figura 11.

Se parte de los clústeres $S_i^{(0)}$ y $Z_{ij}^{(0)}$ expresados en términos de las condiciones iniciales

$$\begin{aligned} S_i^{(0)}\left(x_{i,\partial i}\right) &\equiv p_i^{(0)}\left(x_i\right) \prod_{j\in\partial i} {p_j}^{(0)}\left(x_j\right) \quad \forall i, \\ {Z_{ij}}^{(0)}\left(x_{i,j}\right) &\equiv p_i^{(0)}\left(x_i\right) {p_j}^{(0)}\left(x_j\right) \quad \forall ij. \end{aligned} \tag{C.1}$$

Estos clústeres cumplen la relación de compatibilidad

$$\sum_{x_{\partial i\setminus j}} S_i^{(t)}\left(x_{i,\partial i}\right) = Z_{ij}^{(t)}\left(x_{i,j}\right) \quad \forall i, \forall j \in \partial i \tag{C.2}$$

y permiten calcular las distribución del clúster $P^{(t)}$ a través de la ecuación

$$P_i^{(t)}\left(y_i, x_{i,\partial i}\right) = w_i^{(t)}\left(y_i \mid x_{i,\partial i}\right) S_i^{(t)}\left(x_{i,\partial i}\right) \quad \forall i. \tag{C.3}$$

Marginalizando $P^{(t)}$ se pueden obtener entonces las distribuciones para los clústeres $T^{(t)}$

$$\sum_{x_{\partial i\setminus j}} P_i^{(t)}\left(y_i, x_{i,\partial i}\right) = T_{i,ij}^{(t)}\left(y_i, x_{i,j}\right) \quad \forall i, \forall j \in \partial i, \tag{C.4}$$

y $V^{(t)}$

$$\sum_{x_{\partial i}} P_i^{(t)}\left(y_i, x_{i,\partial i}\right) = V_i^{(t)}\left(y_i, x_i\right) \quad \forall i. \tag{C.5}$$

Recuperando el clúster $Z^{(t)}$ podemos calcular el clúster maximal $Q^{(t)}$ como

$$Q_{ij}^{(t)}\left(y_{i,j}, x_{i,j}\right) = \frac{T_{i,ij}^{(t)}\left(y_i, x_{i,j}\right) T_{j,ij}^{(t)}\left(y_j, x_{i,j}\right)}{Z_{ij}^{(t)}\left(x_{i,j}\right)} \quad \forall ij. \tag{C.6}$$

De la marginalización de $Q^{(t)}$ se obtienen el clúster $U^{(t)}$ mediante la ecuación

$$\sum_{x_j} Q_{ij}^{(t)}\left(y_{i,j}, x_{i,j}\right) = U_{ij,i}^{(t)}\left(y_{i,j}, x_i\right) \quad \forall i, \forall j \in \partial i, \tag{C.7}$$

y el nuevo valor del clúster Z ($Z^{(t+1)}$) usando

$$\sum_{x_{i,j}} Q_{ij}^{(t)}\left(y_{i,j}, x_{i,j}\right) = Z_{ij}^{(t+1)}\left(y_{i,j}\right) \quad \forall ij \tag{C.8}$$

Finalmente, usando $U^{(t)}$ y $V^{(t)}$ se construye el último clúster maximal

$$R_i^{(t)}\left(y_{i,\partial i}, x_i\right) = \frac{\prod_{j\in\partial i} {U_{ij,i}}^{(t)}\left(y_{i,j}, x_i\right)}{V_i^{(t)}\left(y_i, x_i\right)^{d_i-1}} \quad \forall i, \tag{C.9}$$

y con él, se recalcula el valor del clúster S para obtener

$$\sum_{x_i} R_i^{(t)}\left(y_{i,\partial i}, x_i\right) = S_i^{(t+1)}\left(y_{i,\partial i}\right) \quad \forall i. \tag{C.10}$$

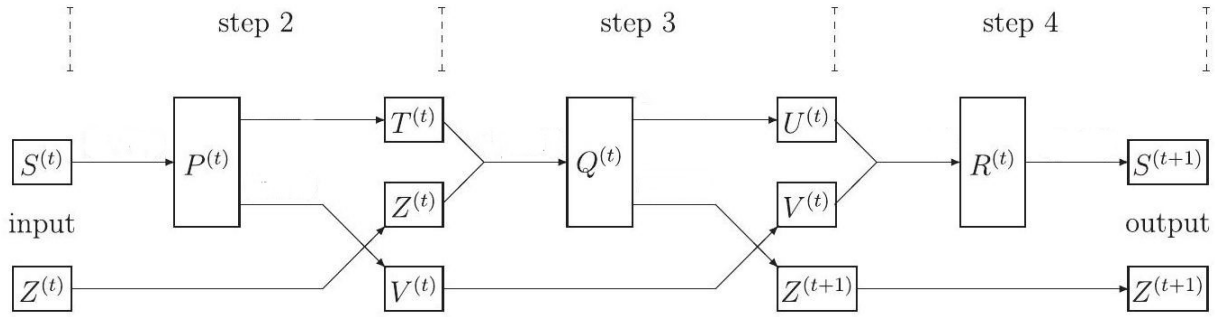

Figura 11: Diagrama de flujo para el cálculo de los distintos clústeres para la aproximación PQR. Las cajas más grandes representan los clústeres maximales, y las más pequeñas los subclústeres. Imagen obtenida de [2].

Como nota adicional, cabe destacar que, si se está interesado en conocer la distribución de probabilidad de un solo nodo en concreto, basta marginalizar el clúster P para obtener

$$A_i^{(t+1)} = \sum_{x_i,\partial i} P_i^{(t)}\left(y_i, x_{i,\partial i}\right) \quad \forall i. \tag{C.11}$$

D. Ampliación del modelo de ciclo celular

Se recogen aquí las reglas para las transiciones entre nodos, así como el grafo asociado para el modelo de ciclo celular (figura 12). Notar que en las reglas aparecen los parámetros $CycD\_del$, Rb_del y $Cdc20\_del$, que si bien se podrían modelizar como nodos, en este caso se tratan como la presencia o ausencia de sustancias que hagan decaer la ciclina D, el complejo Rb y la proteína Cdc20, respectivamente. Para todas las simulaciones, su valor se ha tomado como 0.

En la sección D.1 se recoge la equivalencia entre la representación de los estados como números enteros y números binarios. Asimismo, se muestra la traducción de un estado a una configuración concreta de los nodos.

$$
\begin{aligned}
CycD\colon &\begin{cases} \rho_{act} & = 0 \\ \rho_{dec} & = \rho_r \text{ si } CycD\_del,\ 0 \text{ en otro caso} \end{cases} \\
CycE\colon &\begin{cases} \rho_{act} & = \rho_l \text{ si } (Rb^c \wedge E2F),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (Rb^c \wedge E2F,\ \rho_r \text{ en otro caso} \end{cases} \\
CycA\colon &\begin{cases} \rho_{act} & = \rho_l \text{ si } (Rb^c \wedge Cdc20^c \wedge (UbcH10 \wedge cdh1) \wedge (CycA \vee E2F)),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (Rb^c \wedge Cdc20^c \wedge (UbcH10 \wedge cdh1) \wedge (CycA \vee E2F)),\ \rho_r \text{ en otro caso} \end{cases} \\
CycB\colon &\begin{cases} \rho_{act} & = \rho_l \text{ si } (Cdc20^c \wedge cdh1^c),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (Cdc20^c \wedge cdh1^c),\ \rho_r \text{ en otro caso} \end{cases} \\
Rb\colon &\begin{cases} \rho_{act} & = \rho_r \text{ si } (CycD^c \wedge CycB^c \wedge (p27 \vee (CycA \vee CycE)^c) \wedge Rb\_del^c),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (CycD^c \wedge CycB^c \wedge (p27 \vee (CycA \vee CycE)^c) \wedge Rb\_del^c),\ \rho_r \text{ en otro caso} \end{cases} \\
E2F\colon &\begin{cases} \rho_{act} & = \rho_l \text{ si } (Rb^c \wedge CycB^c \wedge (p27 \vee CycA^c)),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (Rb^c \wedge CycB^c \wedge (p27 \vee CycA^c)),\ \rho_r \text{ en otro caso} \end{cases} \\
p27\colon &\begin{cases} \rho_{act} & = \rho_f \text{ si } (CycD^c \wedge CycB^c \wedge ((CycA \vee CycE)^c \vee (p27 \wedge (CycE \wedge CycA)^c))), \\ & 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (CycD^c \wedge CycB^c \wedge ((CycA \vee CycE)^c \vee (p27 \wedge (CycE \wedge CycA)^c))), \\ & \rho_r \text{ en otro caso} \end{cases} \\
Cdc20\colon &\begin{cases} \rho_{act} & = \rho_l \text{ si } (CycB \wedge Cdc20\_del^c),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (CycB \wedge Cdc20\_del^c),\ \rho_r \text{ en otro caso} \end{cases} \\
UbcH10\colon &\begin{cases} \rho_{act} & = \rho_l \text{ si } (((cdh1 \wedge UbcH10)^c \wedge (CycA \vee CycB)) \\ & \vee (CycA^c \wedge CycB^c \wedge (cdh1^c \vee (Cdc20 \wedge UbcH10)))),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (((cdh1 \wedge UbcH10)^c \wedge (CycA \vee CycB)) \\ & \vee (CycA^c \wedge CycB^c \wedge (cdh1^c \vee (Cdc20\ \wedge UbcH10)))),\ \rho_r \text{ en otro caso} \end{cases} \\
cdh1\colon &\begin{cases} \rho_{act} & = \rho_f \text{ si } (Cdc20 \vee (CycB^c \wedge (CycA^c \vee p27))),\ 0 \text{ en otro caso} \\ \rho_{dec} & = 0 \text{ si } (Cdc20 \vee (CycB^c \wedge (CycA^c \vee p27))),\ \rho_r \text{ en otro caso} \end{cases}
\end{aligned}
\tag{D.1}
$$

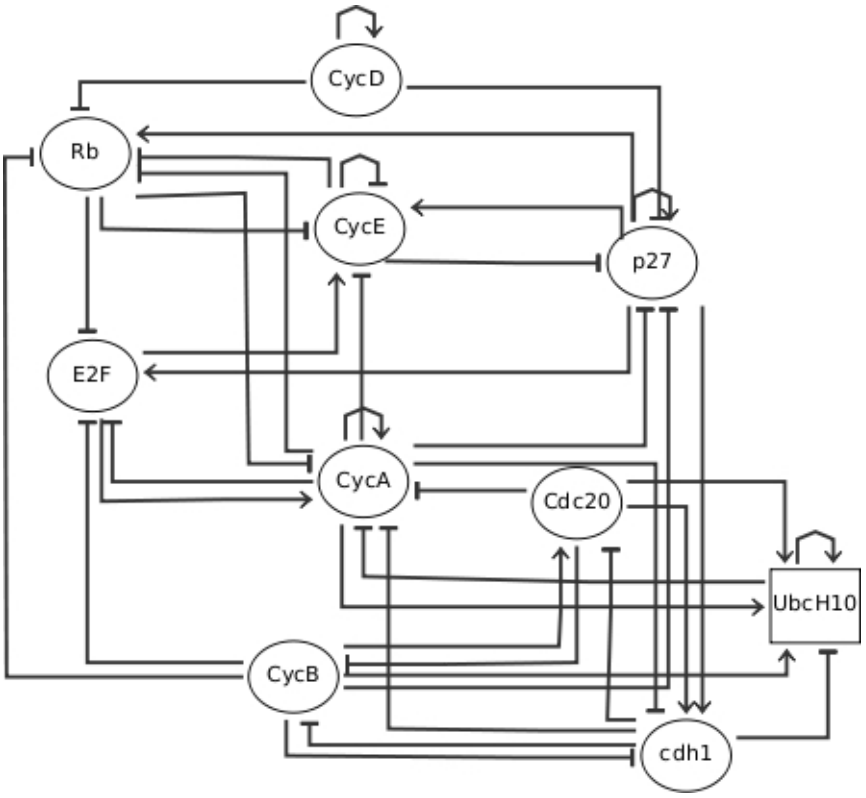

Figura 12: Grafo asociado al modelo de red de regulación del ciclo celular. Imagen obtenida del repositorio web de *MaBoSS*.

D.1. Representación de los estados del ciclo celular

Con el fin de tener una notación más compacta, a lo largo del la sección 3.2 se utilizan números enteros y números binarios para representar configuraciones de los nodos de la red. Puesto que es importante el significado biológico de las especies subyacentes a los nodos, se recoge aquí la forma de traducir los estados entre los distintos enfoques.

Notar, en primer lugar, que el orden en el que se consideran los nodos (que es arbitrario) es el mismo con el que aparecen en la ecuación D.1.

A la hora de describir un estado en concreto en binario, el orden de los digitos respeta el orden de presentación de los nodos. Así, el estado $[x_1\, x_2\, x_3\, x_4\, x_5\, x_6\, x_7\, x_8\, x_9\, x_{10}]$ se correspondería con la configuración

$$\begin{array}{lllll} CycD = x_1, & CycE = x_2, & CycA = x_3, & CycB = x_4, & Rb = x_5, \\ E2F = x_6, & p27 = x_7, & Cdc20 = x_8, & UbcH10 = x_9, & cdh1 = x_{10}. \end{array}$$

Por su parte, cuando se trabaja con los estados representados como números enteros, lo que se hace es invertir el orden de los dígitos de la representación binaria, y traducir ese número a decimal. La razón de proceder así proviene de la elección que se hizo en [3] para el almacenado de los estados.

Como ejemplo, considerar el caso en el que $CycD = 1$ y todos los demás nodos valen 0. En binario, se escribiría [1000000000], y en decimal, sería el estado 1.

E. Repositorio de código

El código empleado a lo largo del proyecto se encuentra almacenado en el repositorio de GitHub

https://gitfront.io/r/DaniUli/r17SsLsdsZ82/CVM/.

Como se ha mencionado antes, en el código se pueden distinguir dos partes diferenciadas, pero complementarias.

Por un lado, se cuenta con un conjunto de ficheros en C que permiten realizar las distintas simulaciones con los tres métodos presentados. La elección de este lenguaje responde principalmente a su velocidad de ejecución.

Por otro lado, se tiene una serie de archivos de Python que permiten procesar una red cualquiera (expresada en un formato .json definido) para obtener los ficheros de entrada de las simulaciones, así como procesar los resultados obtenidos por el programa en C.

Para una mejor comprensión de los archivos empleados, consultar el archivo *README.md* asociado al repositorio.