



Universidad de Zaragoza

Facultad de Ciencias

Departamento de Física de la Materia Condensada

Trabajo de Fin de Grado (Física)

BIOLOGÍA DE SISTEMAS: CIRCUITOS GENÉTICOS OSCILATORIOS Y SINCRONIZACIÓN

Autor:

RAÚL FERNÁNDEZ MELIC

Directores:

DR. FERNANDO FALO

DR. PIERPAOLO BRUSCOLINI

Índice

1. Introducción	2
2. Circuitos genéticos oscilantes:	
Conceptos generales	3
2.1. Transcripción y traducción génica	4
2.2. Factores de transcripción	4
2.3. Reloj/Clock	6
2.3.1. Circuito de realimentación negativa con <i>delay</i>	8
2.3.2. Mecanismo de interacción Delta-Notch	11
2.4. Red con Real. negativa y mecanismo Delta-Notch	13
3. Consideraciones previas a la simulación	15
3.1. Red con Real. Negativa sin interacción Delta-Notch	15
3.2. Parámetro de orden	16
3.3. Discusión sobre los parámetros de las ecuaciones	17
4. Simulaciones y resultados	18
5. Conclusiones	25
A. Funciones de Hill	28
B. Delta-Notch 2 células: Ecuaciones y resultados	30
C. Métodos numéricos	32
D. Valores de parámetros usados en las simulaciones	34
E. Valores de α_{umbral} en función de τ y tamaño de red	36
F. Ruido estático y estocástico	37
F.1. Ruido estático	37
F.2. Ruido estocástico	38
G. Códigos de las simulaciones	40
G.1. Programa principal	40
G.1.1. Cabecera y definiciones	40
G.1.2. Algoritmo de Parisi-Rapuno	41
G.1.3. Runge-Kutta de Orden-4	42
G.1.4. Algoritmo que devuelve posiciones de los vecinos	44
G.1.5. Algoritmos para representación gráfica en GNUPlot	48
G.1.6. Main del programa	49

1. Introducción

Las células de los seres vivos albergan en su interior procesos dinámicos altamente organizados tanto espacial como temporalmente [1] que juegan un papel fundamental en el correcto desarrollo de las funciones biológicas y en la formación de estructuras. El entendimiento preciso del desarrollo de estas dinámicas es fundamental para adquirir una comprensión global de procesos biológicos complejos a escalas mayores que la celular.

Muchos de estos procesos biológicos están asociados a dinámicas oscilantes en la concentración de ciertas especies presentes en el interior de las células. Existen numerosas formas de provocar este tipo de oscilaciones, pero en este trabajo vamos a poner el foco en los osciladores genéticos, cuyo principio de funcionamiento será brevemente descrito en la siguiente sección. Uno de los osciladores genéticos que suscita más interés en la actualidad es el reloj de segmentación, ya que es el responsable del proceso de somitogénesis; se trata de uno de los procesos más importantes en el desarrollo embrionario de los vertebrados y consiste en la segmentación del eje corporal del embrión en somitas, bloques de células epiteliales que en última instancia formarán las costillas, vértebras y músculo esquelético del cuerpo adulto del vertebrado [2]. En 1976, Cooke y Zeeman proponen el famoso modelo "Clock and Wavefront" [3], que presenta al reloj de segmentación como un proceso que consta de dos partes principales:

- En primer lugar depende del denominado "reloj", que consiste en una población de osciladores celulares en fase, es decir, la concentración de una cierta especie a lo largo del tiempo en el interior de una célula de la red oscila en fase con la misma especie del resto de células de la red.
- El segundo ingrediente del reloj de segmentación es un frente de ondas que recorre el embrión y detiene la oscilación de las células, dando lugar a la formación de las somitas.

El embrión cuenta con dos extremos, como se observa en la parte izquierda de la figura 1; el extremo anterior, por el que se van formando las somitas y el extremo posterior, por el que el embrión se elonga [4]. La región en la que se encuentra la población celular oscilando en fase es el "Pre-somitic mesoderm", conocido como PSM. El frente de ondas se desplaza de la parte anterior a la posterior del embrión, congelando a su paso las oscilaciones celulares del PSM y dando lugar a la creación de somitas. De esta forma se traduce la información temporal de los osciladores en un patrón periódico espacial fijo [5], ya que el tamaño de una somita corresponderá a la distancia que recorra el frente de ondas durante un periodo de oscilación del reloj. La parte derecha de la figura 1 muestra de forma visual el mecanismo "Clock and Wavefront".

Además del frente de ondas que viaja de la parte anterior a la posterior del embrión deteniendo las oscilaciones del reloj y formando las somitas, se genera un segundo frente de ondas de expresión génica como consecuencia de las oscilaciones en fase de las células

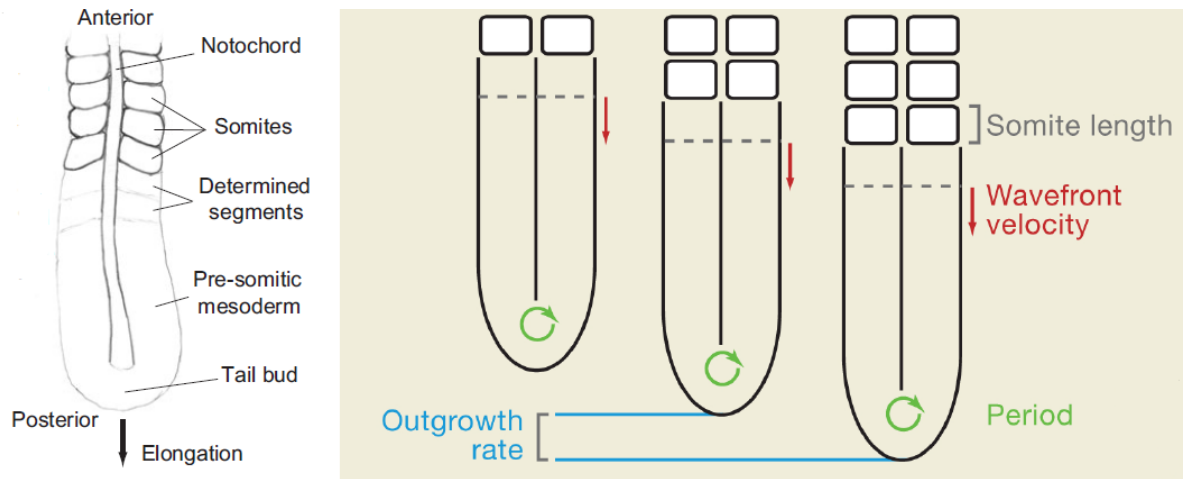


Figura 1: Representación de las partes más relevantes del embrión durante el proceso de somitogénesis (Izquierda)[5] y esquema conceptual del modelo "Clock and Wavefront" (Derecha) [6]

del PSM. Este segundo frente de ondas viaja en sentido opuesto al primero y se detiene en la posición fronteriza de una futura somita [6]. Como ha sido descrito anteriormente, la población celular del PSM ha de presentar oscilaciones sincronizadas, lo que fuerza la presencia de un mecanismo de interacción entre células con el que poder compartir información acerca de la fase de oscilación con el objetivo de acompañarse al ritmo del resto de las células. Existen numerosas evidencias de que los osciladores genéticos del reloj se sincronizan con sus vecinos vía la interacción Delta-Notch [2] [7] [8] [9]. En las páginas sucesivas del trabajo vamos a profundizar en como el reloj (uno de los dos elementos principales que conforman el reloj de segmentación junto al frente de ondas) es capaz de, vía la interacción Delta-Notch, producir oscilaciones sincronizadas en la concentración de una cierta especie presente en el interior celular de todas las células del PSM.

El objetivo último del trabajo es por tanto modelizar matemáticamente un modelo consistente de reloj celular (basándonos en modelos previos), lograr sincronizar una población celular extensa y estudiar las tendencias y particularidades de esta sincronización en función de los parámetros del modelo.

2. Circuitos genéticos oscilantes: Conceptos generales

Para modelizar matemáticamente las oscilaciones en poblaciones celulares y su posible sincronización existen dos aproximaciones:

- Una opción es describir la dinámica del sistema en términos de variables de osciladores genéricos, como la frecuencia o la fase [2]. Esta aproximación se basa en el

modelo de Kuramoto; se trata a cada célula como un oscilador independiente y se modeliza la interacción celular con un término proporcional a la diferencia de fases entre un cierto oscilador y sus vecinos [11]. Así, el cambio de fase:

$$\dot{\theta}_i = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i) \quad (1)$$

siendo ω la frecuencia natural de cada oscilador y K la constante de acoplamiento. Algunos avances importantes en el entendimiento de la somitogénesis se han hecho a través de este acercamiento [12]

- La segunda propuesta pretende que la dinámica del sistema sea descrita en términos de concentraciones e interacciones entre proteínas [2] a través de circuitos genéticos. Como se ha comentado en los párrafos introductorios la interacción entre células viene dada por el mecanismo Delta-Notch.

Este artículo pone el foco en el segundo acercamiento

2.1. Transcripción y traducción génica

Un gen es un segmento de ADN que codifica la información para la creación de una cierta proteína. El proceso de conversión de esa información en una proteína funcional se divide en 2 pasos principales [13]:

- Transcripción: Mediante este proceso se copia la secuencia de ADN de un gen para producir una molécula de ARN-mensajero (ARN-m). Este proceso es catalizado por la ARN-polimerasa (ARN-p) y se inicia con la unión de esta al promotor, que se encuentra al principio de un gen.
- Traducción: Consiste en la conversión de la molécula de ARN-m que contiene la información del gen en una secuencia de aminoácidos que constituye una proteína.

Las proteínas obtenidas mediante este proceso tienen una enorme variedad de funciones, pero hay una pequeña fracción de estas que se unen a fragmentos de ADN y regulan la transcripción. Estas proteínas se denominan factores de transcripción.

2.2. Factores de transcripción

Se conoce como factores de transcripción (FT) a aquellas proteínas que se unen a regiones específicas del ADN (los llamados 'sitios de unión') con el fin de regular la tasa con la que la ARN-p se une al promotor durante la etapa de transcripción. Modifican de esta forma la tasa de producción de la proteína codificada en el gen al que se unen. En función del efecto de la unión de un factor de transcripción a un promotor podemos clasificarlos en dos tipos [14]:

- Conocemos como **activadores** a aquellos factores de transcripción que aumentan la tasa de unión del ARN-p al promotor y por tanto, contribuyen a aumentar la cantidad de ARNm producido. Se incrementa de esta forma la tasa de producción de la proteína codificada en el gen al que se unen.
- Por el contrario, los **represores** son aquellos factores de transcripción que contribuyen a disminuir la cantidad de ARN-p unido al promotor y por tanto, de ARNm producido. Reducen por tanto la tasa de producción de la proteína codificada por el gen al que se encuentran ligados.

Cuando la proteína codificada por un cierto gen activa o reprime la tasa de producción de otra proteína codificada por un segundo gen, se dice que el primer gen activa o reprime al segundo gen, y se representa gráficamente como indican las figuras 2 y 3.

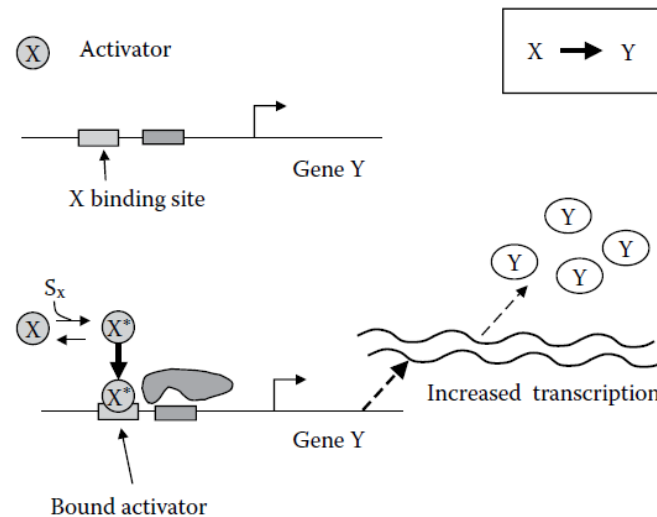


Figura 2: Activación de la proteína de un gen mediante un factor de transcripción activador. Una señal externa hace pasar al factor de transcripción del estado inactivo al activo. En el estado activo, este se une al sitio de unión y produce un aumento en la tasa de proteína producida por el gen. La activación se representa esquemáticamente con una flecha [14].

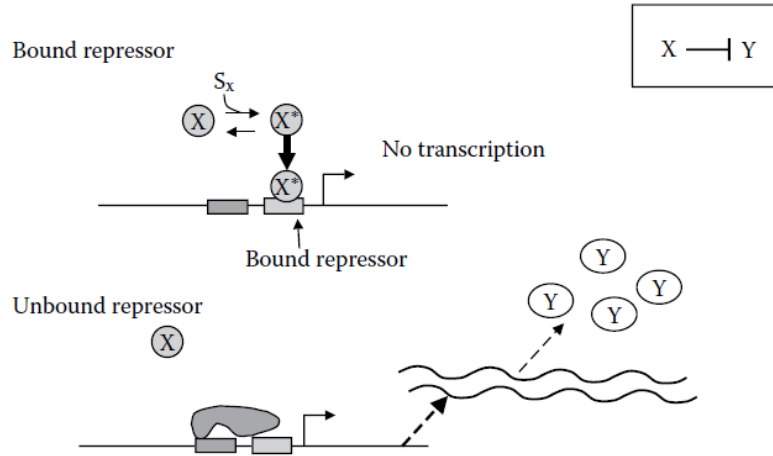


Figura 3: Represión de la proteína de un gen mediante un factor de transcripción represor. Una señal externa induce el estado activo del represor y como consecuencia, este se une al sitio de unión. Esto produce un descenso en la tasa de producción de la proteína codificada por el gen. La represión se representa esquemáticamente con una flecha cortada [14].

Las complejas interacciones biológicas entre proteínas y genes pueden resumirse en redes genéticas compuestas por nodos que representan genes y flechas que los unen, indicando que las proteínas producto de un cierto gen actúan como reguladoras de las proteínas codificadas por el gen al que apuntan. Además, los factores de transcripción pueden regular su propia expresión, lo que da lugar a circuitos genéticos de realimentación positiva o negativa, en función de si la proteína se comporta como un activador o como un represor.

Cuando la solución de las ecuaciones diferenciales que rigen la dinámica de una proteína en el interior de una célula es un ciclo límite, la concentración de esta proteína oscila de forma estable en el tiempo y tenemos por tanto un circuito genético oscilante para los parámetros de las ecuaciones que dan oscilaciones. Estas soluciones son especialmente interesantes, ya que como adelantaba la introducción, la oscilación en fase de la concentración de una cierta proteína de todas las células que forman el PSM del embrión de vertebrados es un requisito imprescindible para que el proceso de somitogénesis se desarrolle de forma exitosa.

2.3. Reloj/Clock

El reloj de segmentación es un circuito de osciladores genéticos celulares que actúa como un sistema de formación de patrones rítmicos a nivel celular [2]. En los párrafos introductorios se ha presentado como la pieza fundamental que nos ayuda a entender como se da la formación de somitas en vertebrados y se han descrito brevemente sus dos partes principales: el reloj y el frente de ondas. Las interacciones genéticas presentes en el reloj para una población de dos células pueden representarse esquemáticamente como

muestra la figura 4.

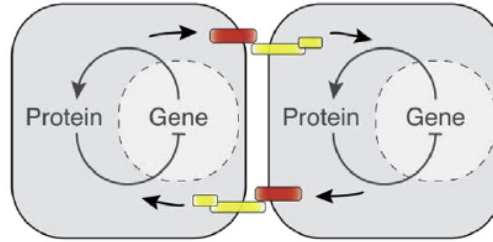


Figura 4: Representación esquemática del reloj en una población de 2 células. La especie roja representa el ligando Delta y la amarilla Notch. En este esquema las flechas negras no representan activación, sino que simplemente muestran la dirección en la que se produce el mecanismo de interacción entre células [2].

Depende esencialmente de 3 especies: El ligando Delta, Notch y una proteína autorreprimida que denominaremos proteína Her/Hes. Podemos dividir el modelo en dos partes diferenciadas:

- En primer lugar, cada célula individual del PSM cuenta con un ciclo de realimentación negativa que da lugar a oscilaciones autosostenidas en la concentración de la especie Her/Hes. La idea de que las células del PSM sean osciladores celulares autosostenidos está ampliamente respaldada [9] [10]. Aunque este circuito da lugar a oscilaciones en todas las células del tejido, estas no están sincronizadas entre distintas células, por lo que se vuelve necesario un método de comunicación/interacción celular.
- La necesidad de un método de interacción celular que pueda dar lugar a sincronización en las fases de los osciladores queda solventada con el mecanismo Delta-Notch, la segunda de las partes fundamentales del reloj.

Con el objetivo de comprobar si este mecanismo es lo suficientemente robusto como para explicar la sincronización en una población celular extensa, es necesario modelizar las ecuaciones diferenciales del sistema y resolverlas numéricamente en busca de soluciones que evidencien una sincronización celular global.

Vamos a comenzar analizando el procedimiento para la obtención de ecuaciones diferenciales que rijan la dinámica de células que presenten sólo interacción Delta-Notch o sólo un circuito de realimentación negativa, con el fin posterior de unir ambos resultados para la descripción de un sistema celular regido por los circuitos genéticos presentes en el reloj.

2.3.1. Circuito de realimentación negativa con *delay*

Vamos a analizar la ecuación diferencial que describe la evolución de la concentración de una proteína que reprime el gen que la codifica (Figura 5).

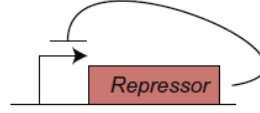


Figura 5: Representación gráfica de un circuito de realimentación negativa [15].

La ecuación diferencial que nos da la evolución temporal de la concentración de la proteína consiste en dos términos, uno de producción y otro de degradación. Generalmente el término de producción se modeliza a través de una función de Hill, que nos da la cantidad de proteína represora producida en función de la concentración de factor de transcripción. Dado que en un circuito de realimentación negativa la proteína producida y el FT son la misma especie, su producción está regulada por sí misma. Además, la proteína está sometida a procesos de degradación, por lo que es necesario modelizar estas pérdidas con un término que tenga en cuenta estos efectos. El término de degradación principalmente engloba dos efectos: La pérdida de cierta especie como resultado de procesos metabólicos y la disminución en la concentración de la especie como consecuencia de un aumento en el volumen celular. En el caso de un circuito de realimentación negativa, la ecuación diferencial que describe la concentración de la proteína puede expresarse [15]:

$$\dot{x} = \frac{\beta}{1 + \left(\frac{x}{K}\right)^n} - \gamma x \quad (2)$$

El parámetro β expresa la máxima tasa de producción de la proteína, γ es la tasa de degradación, K es la concentración de represor necesaria para obtener la mitad de la represión posible y n es el coeficiente de Hill, que indica el índice de cooperatividad del factor de transcripción, es decir, da una medida del número de monómeros que han de unirse para dar lugar a una unión entre el factor de transcripción y el promotor de un cierto gen. Las funciones de Hill tienen comportamiento sigmoideo, para n bajos la pendiente es suave y el comportamiento es más brusco a medida que aumentamos n . Cuando n es muy grande tenemos un comportamiento del tipo función lógica/escalón, es decir, sólo tendremos represión superado un cierto umbral de concentración del FT, dado por la constante K . Para entender con algo más de detalle por qué las funciones de Hill represoras se modelizan matemáticamente de la forma expresada en el término de producción de la ecuación (2) se recomienda la lectura del Apéndice A.

Sin embargo, esta ecuación tiene sólo una dimensión, por lo que desde el punto de vista de los sistemas dinámicos no existe ninguna solución que de lugar a oscilaciones en la

concentración de x . Podemos complicar el modelo de 2 formas para dar lugar a oscilaciones [15]:

- Al modelizar las ecuaciones de esta forma estamos sobresimplificando la dinámica real biológica, en la que los procesos de transcripción y traducción presentan una gran cantidad de especies intermedias entre la unión del FT al promotor y la consecuente disminución de proteína. Una de estas especies es el ARNm, cuya dinámica puede incluirse para añadir una dimensión extra a las ecuaciones y permitir soluciones oscilantes:

$$\dot{m} = \frac{\alpha_m}{1 + \left(\frac{x}{K}\right)^n} - \gamma m \quad (3)$$

$$\dot{x} = \beta m - \gamma x \quad (4)$$

En este caso m representa la concentración de ARNm producido y x la cantidad de proteína represora. El factor β cuantifica la fracción de ARNm que se traduce exitosamente a proteína.

Este proceso de adición de especies intermedias puede iterarse más veces para hacer el modelo más realista con las dinámicas biológicas.

- Aunque el modelo descrito es completamente válido para la obtención de oscilaciones, cada nueva ecuación introducida incrementa el número de parámetros y dificulta la búsqueda de ciclos límites. Una alternativa más sencilla es modelizar las dinámicas intermedias a través de la inclusión de un *delay* en el ciclo de realimentación, es decir, hacemos que la evolución temporal de la concentración \dot{x} dependa de su valor en un tiempo pasado $x_\tau \equiv x(t - \tau)$ en vez de que dependa de su valor actual x . Modelizamos así el hecho de que la realimentación no sea inmediata como consecuencia de las escalas temporales de las especies intermedias mediante la introducción de este *retraso*. Así, la ecuación diferencial para la concentración de una proteína en un circuito de realimentación negativa queda:

$$\dot{x} = \frac{\beta}{1 + \left(\frac{x_\tau}{K}\right)^n} - \gamma x \quad (5)$$

Pero no todos los *delays* producen oscilaciones en la concentración, de hecho, existe un τ_{umbral} por debajo del cuál no tendremos soluciones oscilatorias. La figura 6 muestra las diferentes soluciones de la ecuación diferencial para diferentes valores de τ .

Para *retrasos* menores al valor de τ_{umbral} la concentración alcanza un valor de equilibrio (Figura 6.1). A medida que τ crece (Figuras 6.2 y 6.3) el sistema acaba llegando a un valor constante de concentración, pero precedido por oscilaciones amortiguadas. Cuando sobrepasamos finalmente el valor de τ_{umbral} (Figura 6.4), observamos oscilaciones sostenidas y estables en la concentración de la proteína.

Otra característica curiosa de estos circuitos genéticos es que el valor de *retraso* a partir del cuál vamos a obtener oscilaciones en la concentración de la proteína (τ_{umbral}) va a

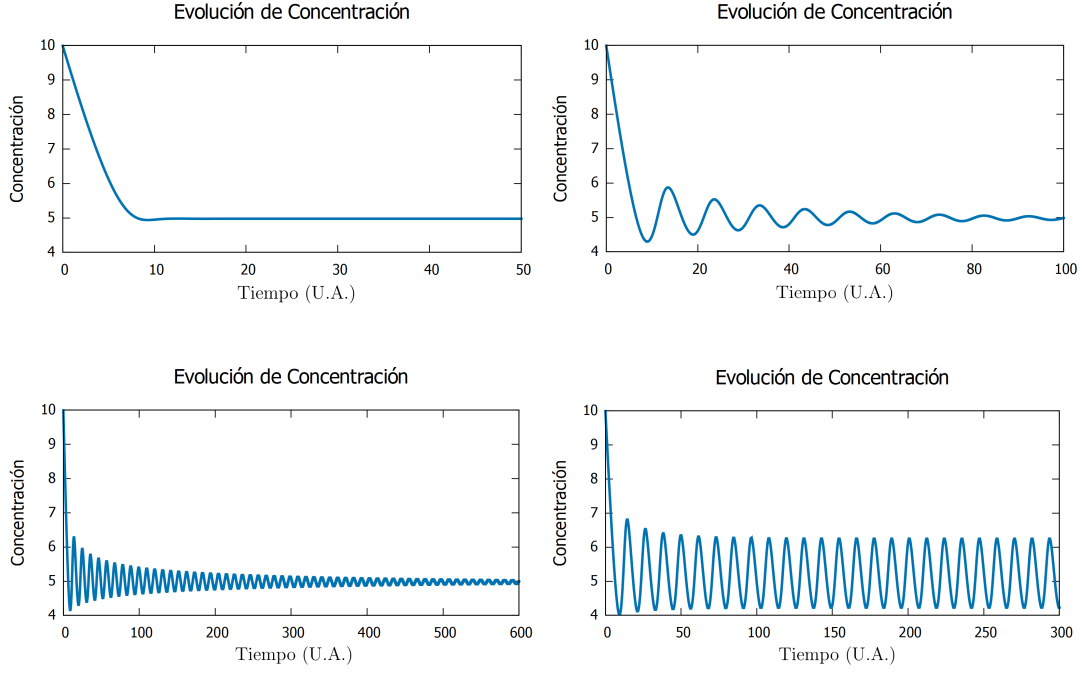


Figura 6: Evolución temporal de la proteína represora en un circuito de realimentación negativa en función del *retraso*. El sistema presenta un coeficiente de Hill $n=4$. Figura 6.1: $\tau=1$ U.A. ; Figura 6.2: $\tau=2.5$ U.A. ; Figura 6.3: $\tau=2.75$ U.A. ; Figura 6.4: $\tau=3$ U.A.

estar fuertemente influenciado por el exponente de Hill n , como puede comprobarse en la figura 7. A medida que crece el índice de cooperatividad se obtiene una rápida caída en el valor de *delay* umbral, por lo que cuánto mayor es n menor *retraso* es requerido para que el sistema alcance oscilaciones estables.

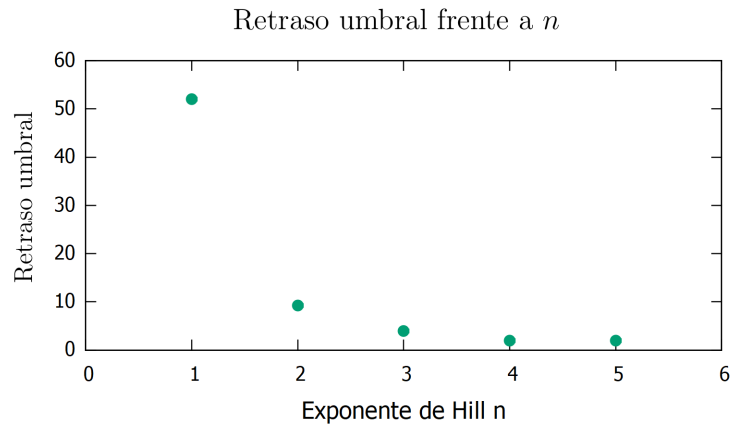


Figura 7: Dependencia de τ_{umbral} con el exponente de Hill n

2.3.2. Mecanismo de interacción Delta-Notch

La sincronización de una población celular requiere que cada célula disponga de un método con el que obtener información acerca de la fase de sus vecinas, es decir, es imprescindible modelizar un método de interacción celular. Como hemos visto anteriormente en el reloj, esta interacción celular viene dada por el mecanismo Delta-Notch, que se basa en 2 sencillas premisas, ilustradas en la figura 8 para el caso de sólo dos células [16]:

- El ligando Delta (d_i) de cada célula activa la producción del represor Notch de las células adyacentes.
- El represor Notch (r_i) de cada célula inhibe la producción de ligando Delta de su misma célula.

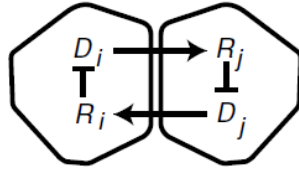


Figura 8: Interacción Delta-Notch en dos células

Para describir este sistema necesitamos por tanto 4 ecuaciones diferenciales, 2 para las especies en la célula 1 y otras 2 para las especies en la célula 2. Las ecuaciones normalizadas que rigen la dinámica de cada célula son las siguientes:

$$\frac{dd_i}{dt} = \nu \left[\frac{\beta_d}{1 + r_i^h} - d_i \right] \quad (6)$$

$$\frac{dr_i}{dt} = \frac{\beta_r d_j^m}{1 + d_j^m} - r_i \quad (7)$$

El parámetro β_d está relacionado con la producción de ligando, β_r con la producción del represor y ν es el cociente de las tasas de degradación ligando-represor.

En el Anexo B se expone en detalle el procedimiento utilizado para llegar a las ecuaciones normalizadas (6) y (7) y se muestran las soluciones obtenidas al resolverlas.

Cuando en vez de con dos células contamos con una red celular extensa, hay que tener en cuenta algunos cambios. Una suposición razonable y común a la hora de describir poblaciones celulares es la de red hexagonal. Las células en esta aproximación se describen como hexágonos formando parte de una red regular, de forma que cada célula individual cuenta con 6 vecinos. Como consecuencia del incremento del número de vecinos y por tanto de las interacciones celulares, las ecuaciones diferenciales cambian. Ahora, el represor de una cierta célula i ya no está activado únicamente por la Delta de su vecino adyacente,

sino que sus 6 vecinos la activan. La ecuación diferencial del represor (7) sufre por tanto la siguiente modificación [16]:

$$\frac{dr_i}{dt} = \frac{\beta_r \langle d_j \rangle^m}{1 + \langle d_j \rangle^m} - r_i \quad (8)$$

donde $\langle d_j \rangle$ representa la media de concentración de ligando de los primeros vecinos de la célula i:

$$\langle d_j \rangle = \frac{1}{W} \sum_{j \in nn(i)} d_j \quad (9)$$

W representa el número de primeros vecinos.

Para que los bordes no supongan un problema de homogeneidad en la red y todas las células cuenten con 6 vecinos se han implementado condiciones de contorno periódicas. Simulando una red cuadrada 12x12 con interacción Delta-Notch entre células vecinas alcanzamos un estado de equilibrio en el que algunas de las células acaban sólo con concentración de Delta y otras sólo con Notch. Si representamos espacialmente las concentraciones de la especie Delta para un tiempo en el que se ha alcanzado el equilibrio (Figura 9), podemos observar la formación de un patrón hexagonal perfecto, en el que por cada célula que tiene sólo concentración de Delta (nodo amarillo), tenemos 2 células con sólo concentración de Notch (nodo negro). Además se observa como cada célula con concentración Delta tiene a sus 6 vecinos más próximos sólo con concentración Notch, lo que da lugar al patrón hexagonal.

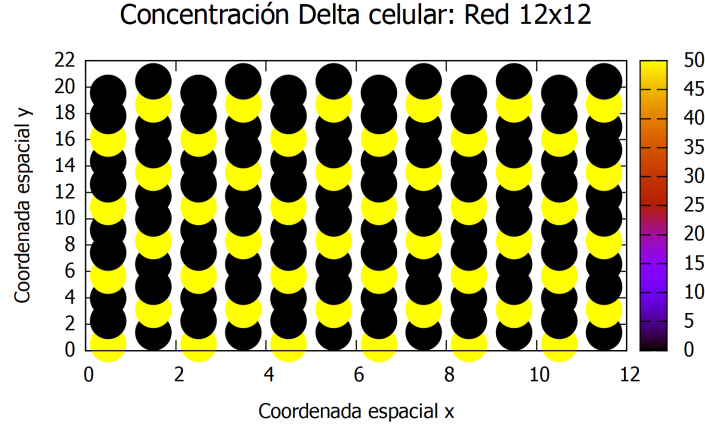


Figura 9: Representación espacial de las concentraciones de equilibrio de la especie Delta en una red hexagonal 12x12 con mecanismo de interacción Delta-Notch

Para la obtención del patrón perfecto se han elegido condiciones iniciales especiales que favorezcan el estado de equilibrio final. Si partimos de concentraciones de Delta y Notch completamente aleatorias para todas las células de la red obtenemos el mismo resultado pero con un patrón defectuoso en el que la estructura hexagonal es menos evidente y en el que dejamos de tener un ratio de nodos amarillos:negros de 1:2 para pasar a tener

ratios más pequeños en los que parte de los nodos que deberían ser amarillos aparecen negros. Conceptualmente sucede algo similar al caso de 2 células. Las diferencias en la concentración tienden a amplificarse y a dejar células sólo con concentración de Delta y otras sólo con Notch. Al hacer la red más grande y proponer que cada célula cuente con 6 vecinos con los que interacciona, el ratio de células que acaban en un estado o en otro pasa de ser 1:1 (propio de redes lineales con interacciones en una dimensión) a ser 1:2 como consecuencia de modelizar una red hexagonal con interacciones en dos dimensiones.

2.4. Red con Real. negativa y mecanismo Delta-Notch

Una vez han sido descritas las formas de modelizar las dos partes principales del reloj de forma individual vamos a obtener las ecuaciones diferenciales resultantes de combinar estos dos efectos. El circuito genético que queremos describir puede visualizarse en la figura 11, aunque en nuestro caso no va a restringirse a la interacción entre 2 células sino a toda una población celular. Ya hemos visto que cada célula del modelo cuenta con 3 especies en su interior, el ligando Delta, que activa el Notch de otras células; la especie Notch, que activa la proteína Her/Hes de su misma célula y la proteína Her/Hes, que reprime al ligando Delta de su misma célula y a si mismo. El ligando Delta y Notch se modelizan a través de las ecuaciones (6) y (7) respectivamente (se ha cambiado ligeramente la notación para hacer explícito que las nuevas ecuaciones describen al reloj celular y no al mecanismo Delta-Notch, ya que para estas dos especies las ecuaciones resultantes al describir el reloj no se modifican respecto a las vistas en el apartado anterior), pero la proteína Her/Hes en el circuito celular no está únicamente autorreprimida sino que también está activada por Notch. Para describir matemáticamente esta doble interacción vamos a hacer uso de la solución propuesta por Hasty et al. [17] para resolver el problema del oscilador con realimentación dual (ORD), expuesto en la figura 10. Este circuito está compuesto por dos genes, denominados *araC* y *lacI*. El primero de ellos está activado por sí mismo y reprimido por *lacI* mientras que el segundo está activado por *araC* y reprimido por sí mismo. La situación del gen *lacI* en el circuito ORD es equivalente a la del gen *Her/Hes* en el reloj celular, por lo que las ecuaciones que describen la evolución de la proteína *lacI* en el ORD son a su vez válidas para *Her/Hes*.

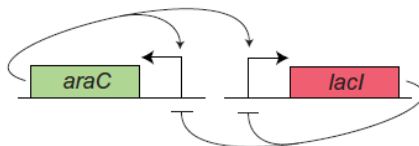


Figura 10: Esquema del oscilador con realimentación dual [15].

Las ecuaciones del sistema pueden describirse por tanto de la siguiente forma:

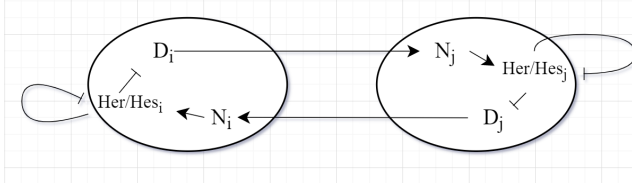


Figura 11: Modelo de 3 ecuaciones

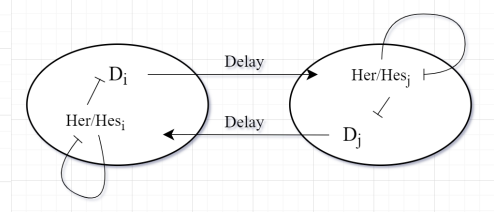


Figura 12: Modelo de 2 ecuaciones

Figura 13: Comparación entre el circuito genético que considera la dinámica de Notch y el circuito genético que reemplaza a la especie Notch por un *delay*.

$$\text{Ligando Delta : } \frac{dD_i}{dt} = \nu \left[\frac{\beta_d}{1 + H_i^h} - D_i \right] \quad (10)$$

$$\text{Notch : } \frac{dN_i}{dt} = \frac{\beta_N \langle D_j \rangle^m}{1 + \langle D_j \rangle^m} - N_i \quad (11)$$

$$\text{Proteína Her/Hes : } \frac{dH_i}{dt} = p_H \frac{1 + \alpha N_i^2}{[1 + N_i^2] [1 + H_{i\tau}^4]} - \gamma_H H_i \quad (12)$$

En la ecuación (12) N_i representa la concentración de Notch que activa la proteína Her/Hes (H_i), α representa el fold-change de la activación y $H_{i\tau} \equiv H_i(t - \tau)$ se refiere a la concentración de la proteína en un tiempo τ anterior, como forma de modelizar el *retraso* inherente a los ciclos de realimentación cuando se tiene en cuenta la dinámica del ARNm.

Aunque estas ecuaciones son perfectamente válidas para buscar sincronización en la red, va a resultar conveniente introducir una modificación que reducirá considerablemente la complejidad del sistema y el número de parámetros implicados.

El ligando Delta de una cierta célula i activa el Notch de la célula vecina j , y a su vez Notch activa la proteína Her/Hes de la célula j . El efecto total de estas dos interacciones en cadena es la activación de la proteína Her/Hes a partir del ligando Delta, por lo que si modelizamos esa doble activación ($D_i \rightarrow N_j \rightarrow Her/Hes_j$) como una sola activación ($D_i \rightarrow Her/Hes_j$) podemos prescindir de la especie Notch y simplificar por tanto enormemente el sistema. Sin embargo, cuando simplificamos un paso del circuito genético, indirectamente estamos atribuyendo una inmediatez en la creación de la proteína Her/Hes ficticia, ya que la especie Notch introduce un *retraso* en el sistema de forma natural. Para que la "eliminación" de la especie Notch no haga menos realista el modelo, se propone introducir un segundo *delay* τ_2 en las ecuaciones (A partir de ahora el retraso en la autorrepresión de Her/Hes τ pasa a llamarse τ_1 para hacer a los *delays* distinguibles). De esta forma, el *retraso* natural que es inducido en el sistema en el paso ($D_i \rightarrow N_j \rightarrow Her/Hes_j$) mediante una especie Notch es sustituido por un *delay* en la activación simple ($D_i \rightarrow Her/Hes_j$). La figura 13 resume los circuitos genéticos antes y después de la modificación.

Tras la modificación las ecuaciones diferenciales adquieren la siguiente forma:

$$\text{Ligando Delta : } \frac{dD_i}{dt} = \nu \left[\frac{\beta_d}{1 + H_i^h} - D_i \right] \quad (13)$$

$$\text{Proteína Her/Hes : } \frac{dH_i}{dt} = p_H \frac{1 + \alpha \langle D_{j\tau_2} \rangle^2}{[1 + \langle D_{j\tau_2} \rangle^2] [1 + H_{i\tau_1}^4]} - \gamma_H H_i \quad (14)$$

En la ecuación (14), $\langle D_{j\tau_2} \rangle \equiv \langle D_j(t - \tau_2) \rangle$ sustituye a lo que anteriormente era N_i . Con la modificación, ahora el término de producción de la proteína Her/Hes depende directamente de la concentración de ligando Delta, que además es la concentración del ligando en un tiempo τ_2 anterior como consecuencia de la eliminación de la especie Notch. Los *delays* τ_1 de la autorrepresión de la proteína Her/Hes y τ_2 de la activación del ligando Delta a Her/Hes se han supuesto diferentes en las ecuaciones porque la escala temporal del bucle de realimentación y del mecanismo Delta-Notch no tiene por que ser la misma; sin embargo, serán considerados equivalentes ($\tau_1 \equiv \tau_2 \equiv \tau$) en las simulaciones por sencillez.

3. Consideraciones previas a la simulación

3.1. Red con Real. Negativa sin interacción Delta-Notch

Antes de comenzar la resolución numérica de las ecuaciones (13) y (14) tenemos que determinar los valores de los parámetros que en ausencia de interacción Delta-Notch dan lugar a oscilaciones, ya que para la obtención de sincronización es necesario que la concentración de proteína Her/Hes de cada célula oscile independientemente al resto. Además, dado que los objetivos del trabajo giran entorno a la obtención de sincronización, es especialmente importante comprobar que el barrido en condiciones iniciales es lo suficientemente eficaz como para asegurar que la sincronización obtenida se basa en los mecanismos de interacción celular y no en la proximidad de los parámetros en las condiciones iniciales de las diferentes células. Para solventar ambos problemas es conveniente generar una red en la que estén modelizados los ciclos de realimentación negativa en cada célula pero no los mecanismos de interacción celular Delta-Notch. En nuestras ecuaciones esto se consigue haciendo $\alpha=1$, ya que con ese valor la ecuación diferencial de la proteína Her/Hes elimina la dependencia de parámetros de células vecinas ($D_{j\tau}$) y se recupera la forma de un circuito de realimentación negativa:

$$\frac{dH_i}{dt} = \frac{p_H}{1 + H_{i\tau}^4} - \gamma_H H_i \equiv \frac{dx}{dt} = \frac{\beta}{1 + \left(\frac{x_\tau}{K}\right)^n} - \gamma x \quad (15)$$

Por su parte, la ecuación diferencial del ligando Delta se mantiene invariable al no depender de parámetros de células vecinas.

Con los parámetros adecuados podremos obtener oscilaciones en la concentración de cada proteína individual y se deberá verificar que este grupo de osciladores independiente

presenta homogeneidad y dispersión en las fases. De esta forma, cuando integremos las ecuaciones del reloj y encontremos sincronización (osciladores en fase) podremos estar seguros de que los responsables han sido los mecanismos de interacción celular. La figura 14 ha sido obtenida en una red integrada por las ecuaciones (13) y (15) (sin interacción Delta-Notch) y confirma la necesaria homogeneidad en las fases de los osciladores independientes.

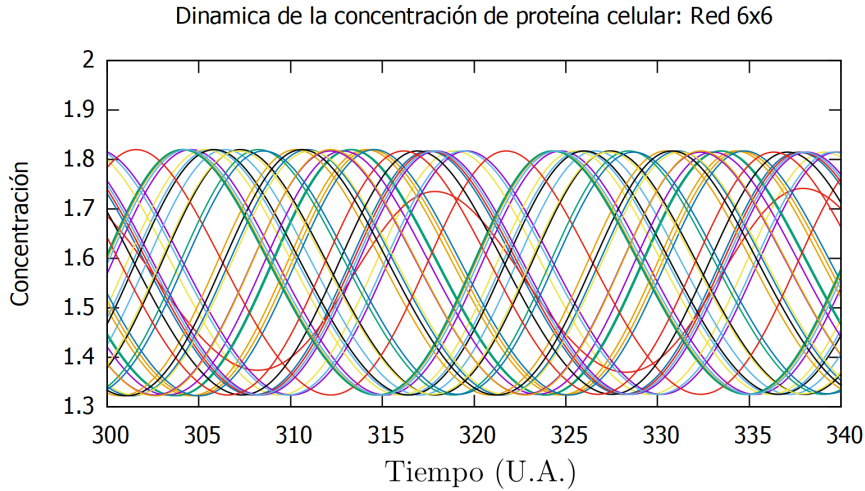


Figura 14: Concentración en función del tiempo de las proteínas Her/Hes de cada una de las 36 células de una red 6x6 en ausencia de interacción celular

3.2. Parámetro de orden

Para evaluar de forma cuantitativa la posible sincronización en nuestro sistema celular es indispensable definir un parámetro de orden normalizado cuyo valor sea nulo en ausencia total de sincronización y que valga la unidad cuando las concentraciones de proteína Her/Hes de cada célula de la red oscilen completamente en fase. Un parámetro de orden válido es el propuesto por García-Ojalvo, Elowitz y Strogatz en su intento por acoplar circuitos genéticos represiladores mediante "quorum sensing" [18]:

$$R = \frac{\langle M(t)^2 \rangle - \langle M(t) \rangle^2}{\langle b_i^2 \rangle - \langle b_i \rangle^2} \quad (16)$$

En esta ecuación b_i representa la concentración en un cierto instante de tiempo de la célula i -ésima, $\langle \dots \rangle$ hace referencia a un promedio temporal y $\overline{\dots}$ a un promedio a lo largo de todas las células de la red. Además, se ha definido la señal media $M(t)$ como:

$$M(t) = \frac{1}{N} \sum_{i=0}^N b_i(t) \quad (17)$$

R es por tanto un cociente entre la desviación estándar de la señal promedio y la desviación estándar de b_i promediada sobre las células i .

De esta forma, cuando la red exhiba sincronización, las concentraciones de diferentes células i serán las mismas en cada paso de tiempo e iguales a la concentración promedio $M(t)$ por lo que el cociente será exactamente 1. Cuando por el contrario la red no presente sincronización, cada célula individual presentará una fase independiente del resto de células, por lo que cuando N sea grande tendremos un valor de $M(t)$ aproximadamente constante, haciendo que su desviación estándar sea muy cercana a 0.

3.3. Discusión sobre los parámetros de las ecuaciones

Se han discutido con anterioridad los motivos por los que las ecuaciones (13) y (14) representan una buena aproximación al reloj celular. Estas ecuaciones presentan una buena cantidad de parámetros a determinar, pero no todos ellos son relevantes; nos interesa evaluar la sincronización en función de aquellos parámetros relacionados con la interacción celular. El resto de parámetros presentarán un valor constante a lo largo de todas las simulaciones. Recordando lo visto en la sección 3.1, para lograr sincronización en la red estos parámetros deben permitir oscilaciones independientes en la concentración de la proteína de cada célula en ausencia de interacción celular ($\alpha=1$), por lo que una elección válida es hacer uso de los parámetros utilizados para la obtención de la figura 14 (Apéndice D).

Tenemos por tanto sólo 2 variables que están relacionadas con el mecanismo Delta-Notch simplificado propuesto en la figura 12:

- El parámetro α , que representa el Fold Change de la activación de la proteína Her/Hes por el ligando Delta, definido como:

$$\alpha = \frac{p_{bound}(A \neq 0)}{p_{bound}(A = 0)} \quad (18)$$

Este cociente relaciona la probabilidad de unión del ARNp al promotor (señal de iniciación de la transcripción) cuando la cantidad de activador es no nula frente a esta misma probabilidad en ausencia de activador. Como la presencia del activador aumenta la probabilidad de unión de estas dos especies, en sistemas biológicos reales esta cantidad será siempre mayor que la unidad.

- En segundo lugar, el *delay* τ presente en $D_{j\tau}$ también está relacionado con el mecanismo de interacción celular, ya que como se comentaba en la sección (2.4), este *delay* se introduce con el objetivo de modelizar de forma simplificada la dinámica de la especie Notch, por lo que su valor guarda relación con la escala temporal del mecanismo Delta-Notch.

En las simulaciones vamos a estudiar por tanto la sincronización global R en función de diferentes valores de α y τ . También consideraremos diferentes tamaños de redes celu-

lares cuadradas.

4. Simulaciones y resultados

En una población celular grande es común encontrar ligeras variaciones morfológicas y composicionales entre células. Estas pequeñas diferencias hacen que las magnitudes físicas que definen el estado de las células varíen algo entre ellas. Para hacer el modelo algo más realista se ha introducido en las simulaciones un pequeño ruido estático en los parámetros p_H y β_d , es decir, introducimos aleatoriamente en cada célula pequeñas desviaciones respecto a los valores previamente fijados con el objetivo de modelizar esta inhomogeneidad intrínseca a los sistemas biológicos.

Vamos a comenzar mostrando una serie de gráficas en las que se representa el valor del parámetro de orden R en función de α para diferentes valores de *delay* τ . La figura 15 representa una red celular 6x6. Para verificar si la sincronización es robusta cada uno de los valores de $R_{promedio}$ mostrados en la gráfica son en realidad el resultado de un doble promediado: Temporal (Se obtiene el valor de R en 3000 pasos de tiempo distintos y se calcula su promedio) y en condiciones iniciales (Se calcula el promedio temporal para 200 condiciones iniciales diferentes y el cálculo de ese valor medio genera el $R_{promedio}$ mostrado en la gráfica).

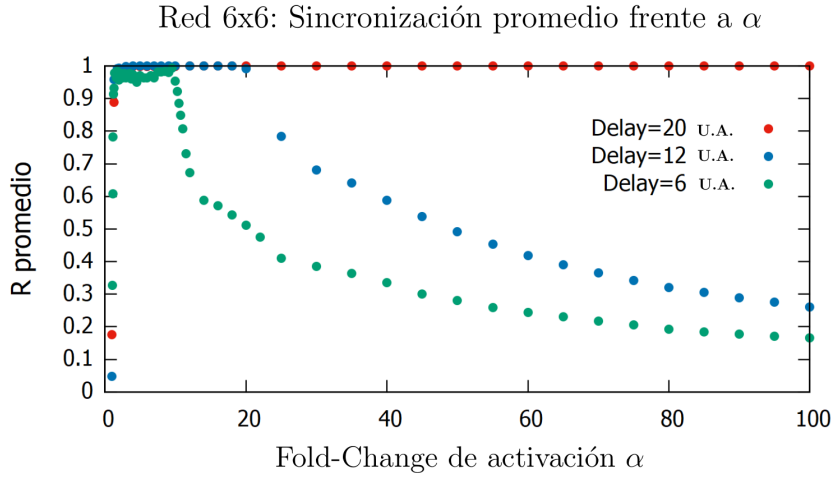


Figura 15: Red 6x6: Sincronización R frente a α para diferentes valores de *delay* τ

Analizando la imagen podemos concluir dos cosas:

- En primer lugar observamos que para los 3 valores de *delay* propuestos encontramos regiones de α para las que se obtiene una sincronización estable y robusta ($R \approx 1$).
- El rango de valores de α que permiten una sincronización robusta depende críticamente del *delay* introducido.

El comportamiento cualitativo de R vs α para un *delay* de 20 U.A. es equivalente a lo que vemos para $\tau=6$ y $\tau=12$, solo que el rango de α propuesto en la gráfica no es suficientemente grande como para visualizar su caída. Todos los *delays* siguen por tanto la misma tendencia: Parten de una desincronización completa en $\alpha=1$ (Para este valor de α no hay interacción celular) y a medida que vamos subiendo α alcanzan muy rápidamente la sincronización global en la red. Se mantienen sincronizados a lo largo de un rango de valores de α y cuando alcanzan el α_{umbral} (valor de α a partir del cuál R deja de ser la unidad) van perdiendo sincronización de forma paulatina hasta alcanzar valores de R cercanos a 0.

Los valores de α_{umbral} para *delays* de $\tau=20$ U.A. y superiores se muestran en el apéndice E. Es importante comentar que la sincronización y en general cualquier estado de equilibrio no se alcanza de forma inmediata sino que se requiere un cierto tiempo para pasar de las condiciones iniciales a los valores de equilibrio. Para obtener valores de R promedio cercanos a 1 se ha de termalizar el sistema antes de iniciar el cálculo de R para dar tiempo a que el sistema alcance el estado globalmente sincronizado.

Veamos con un poco más de detalle el comportamiento de la red en cada una de las fases que acabamos de describir. La figura 16 muestra una serie de gráficas con la evolución de la concentración de la proteína Her/Hes en función del tiempo para cada una de las 36 células que componen la red. Cada gráfica muestra el estado del sistema para un valor de α diferente.

En primer lugar, en el caso de $\alpha=1$ vemos como, al no haber mecanismo de interacción celular la sincronización no es posible y por tanto las fases de cada célula oscilan de forma independiente al resto. Podemos visualizar pequeñas variaciones en las amplitudes de oscilación de diferentes células como consecuencia de la inclusión de ruido estático. La segunda figura ha sido simulada para un valor de α para el que encontramos sincronización global. Como consecuencia, las concentraciones aparecen completamente en fase. Pasado el α_{umbral} , la sincronización promedio empieza a caer lentamente. Sería razonable pensar que sólo existen dos estados extremos posibles en el sistema; el sincronizado global ($R=1$) y el desincronizado global ($R=0$) y que cuando aumenta α se favorecen los estados desincronizados, haciendo descender paulatinamente el promedio de R . Sin embargo, los estados que se obtienen son realmente estados intermedios no completamente sincronizados ni completamente desincronizados. Como se observa en los histogramas de la figura 17, para un determinado α (superior al α_{umbral}) los valores de sincronización R alcanzados en las diferentes simulaciones tienden a un rango acotado de valores (ya no obtenemos estados con sincronización global completa) y a medida que se incrementa el valor de α este rango de valores de R se va acercando de forma gradual a 0.

Desde una perspectiva de concentraciones celulares, la pérdida de sincronización global en la red se traduce en lo que se observa en las gráficas tercera, cuarta y quinta de la figura 16. Para valores cercanos pero mayores a α_{umbral} empieza a advertirse la formación de subgrupos de oscilación en la red. Cada subgrupo oscila completamente en fase con

los osciladores del mismo grupo pero las amplitudes de oscilación y las fases de oscilación entre diferentes grupos comienza a separarse lentamente, lo que da lugar a una pequeña pérdida de sincronización global. A medida que α se aleja más del α_{umbral} los subgrupos de oscilación presentan patrones de oscilación más complejos y las diferencias en amplitud y fase se hacen más evidentes.

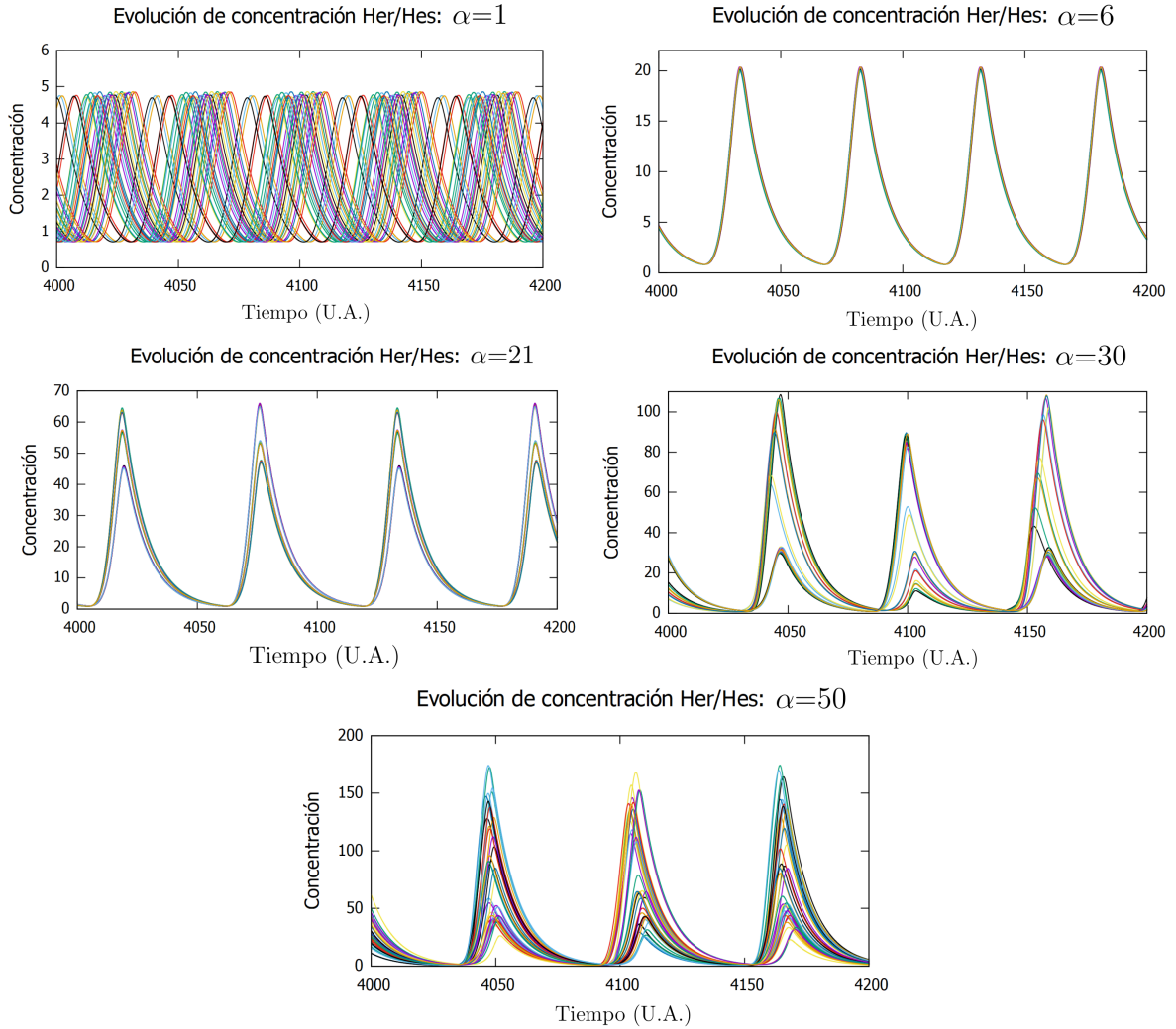


Figura 16: Evolución temporal de la concentración de las 36 células de una red 6x6 para diferentes valores de α . Las simulaciones se han ejecutado con un *delay* $\tau=12$ U.A. Figura 16.1: $R=0.05$; Figura 16.2: $R=1$; Figura 16.3: $R=0.92$; Figura 16.4: $R=0.68$; Figura 16.5: $R=0.49$

El mayor desfase entre subgrupos es el responsable de que el valor de R promedio descienda de forma monótona con el incremento de α . Eventualmente para valores de α elevados se pierde la estructura de subgrupos y las concentraciones celulares oscilan de nuevo de forma casi independiente entre ellas.

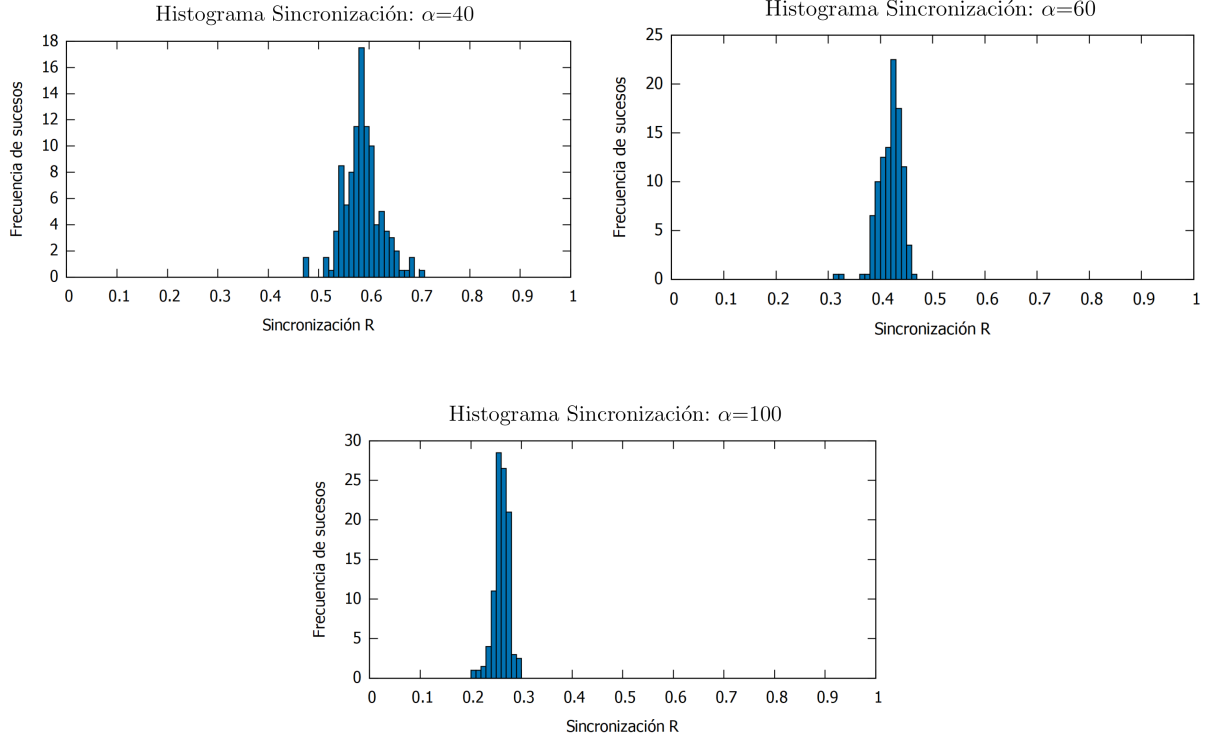


Figura 17: Histogramas de distribución de la sincronización R de los estados de equilibrio para diferentes valores de α superiores al α_{umbral} . Se ha utilizado para la simulación una red 6x6 con $\text{delay } \tau=12$ U.A.

La figura 18 sirve como ejemplo para visualizar la disposición espacial dinámica de la concentración de Her/Hes adoptada por una red 12x12 cuando se supera ligeramente el α_{umbral} ¹. Se puede observar con claridad la separación del sistema en subgrupos de oscilación con diferentes fases.

Para este valor concreto de α se consiguen dos subgrupos independientes, uno de ellos conformado por dos tercios de la red y el otro por el tercio restante. Las fases son claramente diferentes, pero además las amplitudes máximas alcanzadas por cada subgrupo también lo son. Esto se observa más claramente en la figura 19, en la que se representa la concentración de Her/Hes del mismo sistema celular de forma alternativa.

¹Para los parámetros usados en la figura 18 ($\tau=6$ U.A. y red 12x12), estrictamente no se tiene un α_{umbral} ya que la sincronización promedio nunca alcanza $R \approx 1$; sin embargo, los estados de sincronización intermedia que se obtienen tras pasar el valor de α en el que R promedio es máximo ($R \approx 0.6$) son cualitativamente equivalentes a lo que se pretende mostrar y menos costosos de generar computacionalmente, lo que justifica la decisión.

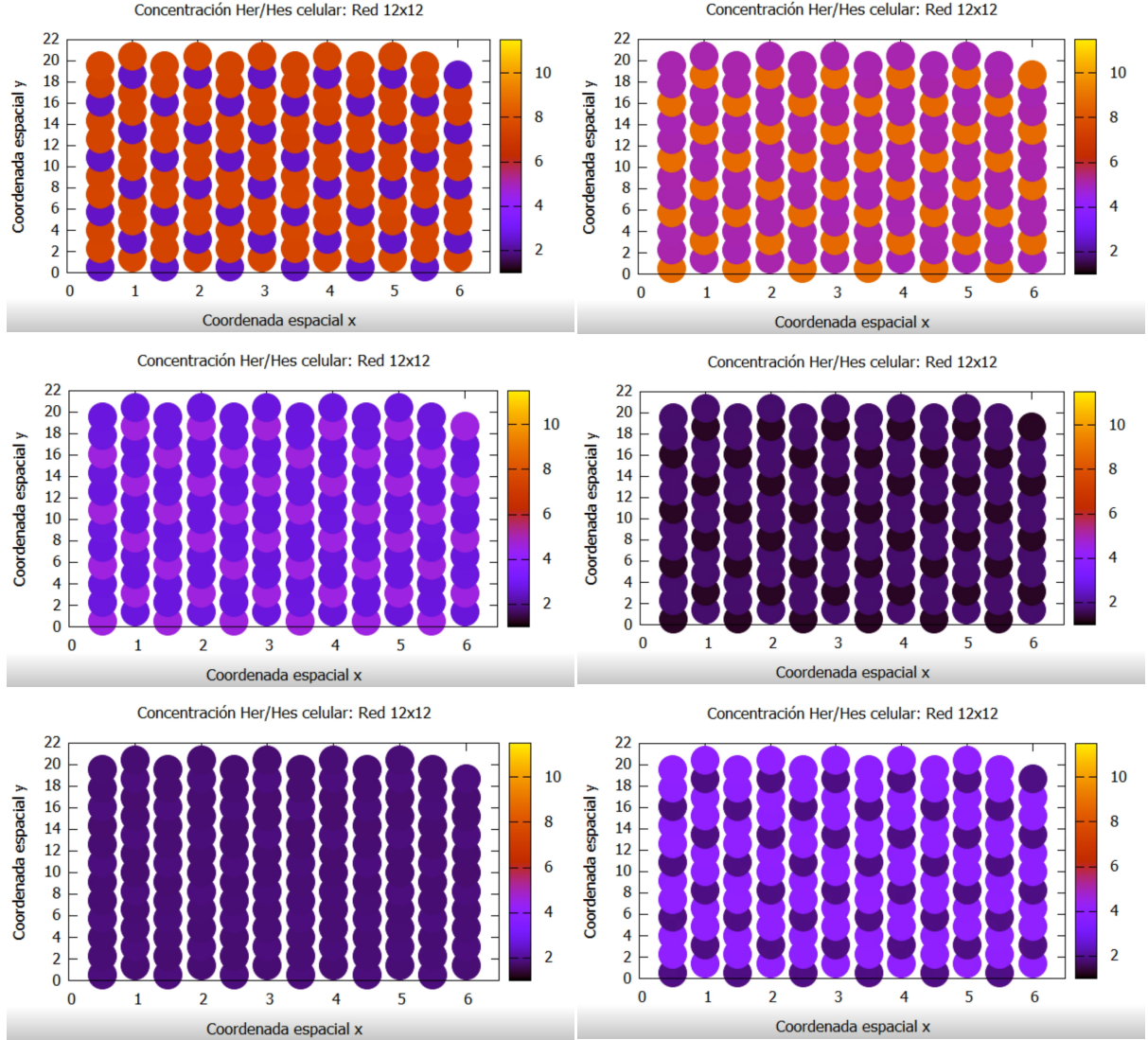


Figura 18: Representación espacial de la evolución de la concentración de las 144 células de una red 12x12 para un α ligeramente superior al α_{umbral} . Las 6 figuras describen diferentes fases del ciclo de oscilación de Her/Hes representado en la figura 19. Se ha usado un valor de $\tau=6$ U.A. y un $\alpha=12$, aunque para valores superiores de τ el comportamiento cualitativo es el mismo.

La riqueza en patrones de oscilación espaciales es grande y para diferentes valores de α obtenemos diferente número de subgrupos de oscilación y con diferentes amplitudes, como se adelantaba en la presentación de la figura 16.

Hasta ahora se han evaluado las tendencias que se pueden extraer del sistema cuando se varía el *retraso* para un mismo tamaño de red. Para completar la discusión es preciso analizar si las tendencias observadas al modificar τ son extrapolables a tamaños mayores de red (Figura 20).

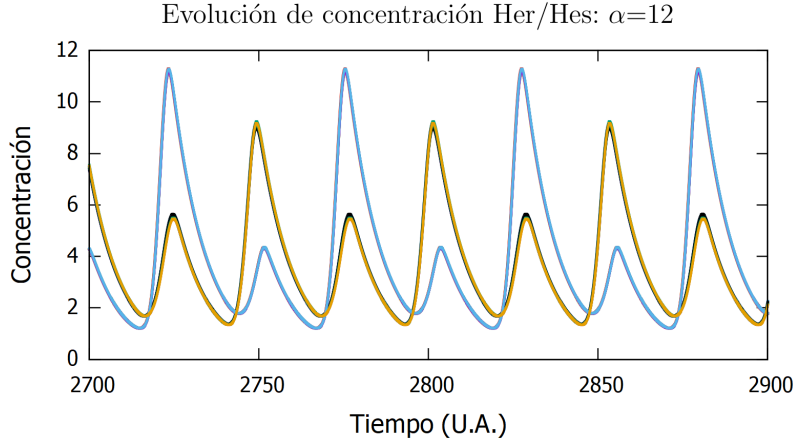


Figura 19: Evolución de la concentración de Her/Hes de las 36 primeras células de una red 12x12 para un $\alpha=12$ (ligeramente superior al α_{umbral}).

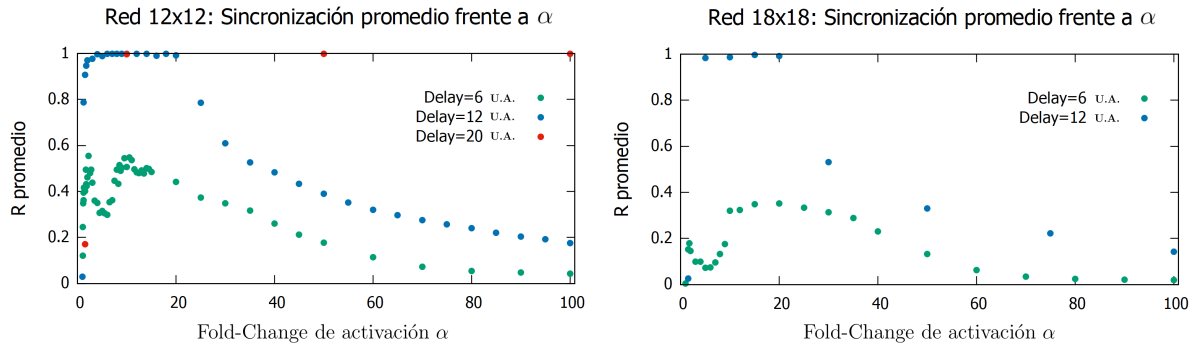


Figura 20: Sincronización R frente a α para diferentes valores de *delay* τ : Redes 12x12 y 18x18

Observamos diferencias notables en las gráficas como consecuencia de aumentar el tamaño de la red, pero esto será analizado más en detalle posteriormente; nos centramos ahora únicamente en los cambios generados por la modificación de τ . Dejando de lado efectos del tamaño de red, tanto para la red 12x12 como para la 18x18 el efecto de aumentar el *delay* en las ecuaciones es comparable con lo obtenido en la red 6x6: En los 3 casos aumentar el *retraso* supone una ampliación en el rango de valores de α para los que se obtiene sincronización global, por lo que se puede concluir que independientemente del tamaño de la red un aumento en el valor de τ supone una mejora notable en la obtención de sincronización. Una vez caracterizado el papel de τ en la inducción de sincronización, es conveniente reorganizar algunos de los resultados presentados para evaluar el efecto de un aumento del tamaño de la población celular. Vamos a estudiar de nuevo como depende la sincronización R del parámetro α , pero en este caso dejaremos fijo el valor de *delay* τ y analizaremos los cambios que induce en el sistema un aumento en el tamaño de la red. La figura 21 muestra los resultados:

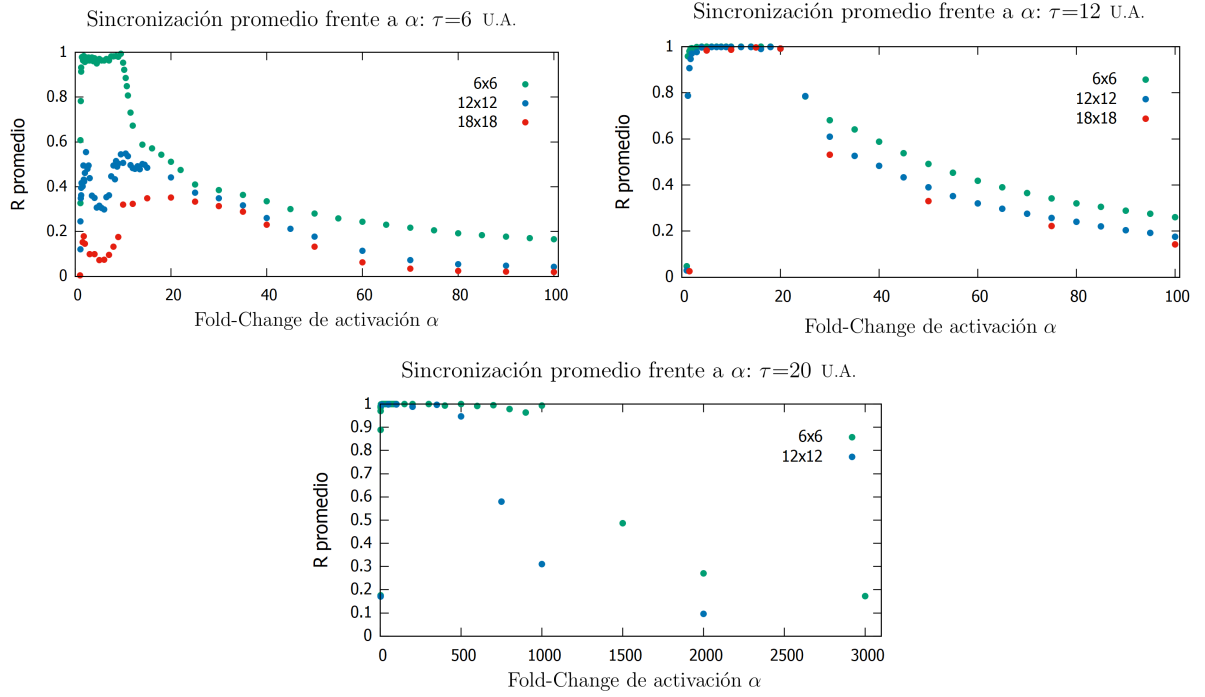


Figura 21: Sincronización R frente a α para diferentes valores de tamaño de red: *delays* de 6, 12 y 20 U.A.

Observando las figuras podemos extraer varias conclusiones:

- En primer lugar, cuando las ecuaciones presentan un *delay* $\tau=6$ U.A. se observa claramente como el aumento del tamaño de la red se traduce en una pérdida clara de sincronización promedio. Además, también se puede deducir que un *retraso* de 6 U.A. es probablemente demasiado pequeño para inducir sincronización robusta en redes mayores que la 6x6.

A diferencia de lo que sucede para valores de α superiores al α_{umbral} , cuando para α bajos obtenemos R promedio menores de 1, es como consecuencia de obtener el promedio entre estados sincronizados ($R=1$) y no sincronizados ($R=0$), de forma que un $R=0.6$ indica que aproximadamente el 60 % de las simulaciones han alcanzado el estado sincronizado mientras que el resto no ha conseguido sincronizar.

- Para *delays* $\tau=12$ U.A. la influencia de aumentar el tamaño de la red es menos distinguible. En este caso todos los tamaños presentan aproximadamente el mismo valor de α_{umbral} . Para valores superiores a α_{umbral} si que se observa un valor de sincronización menor a medida que aumenta el tamaño de red.

Para *delays* $\tau=20$ U.A. el aumento de la red se traduce en un α_{umbral} menor. Es importante notar que un incremento pequeño en τ aumenta de forma muy notable los valores de α_{umbral} , como se observa en el apéndice E.

En general se puede concluir que un aumento en el tamaño de la red genera mayores dificultades para sincronizarla. Como es lógico cuánto mayor es la red, mayor es el número

de células y de interacciones celulares y por tanto más difícil conseguir que las concentraciones de proteína Her/Hes de las células oscilen en fase. Esta mayor dificultad se hace especialmente evidente en el caso de $\tau=6$ U.A., al no ser este valor lo suficientemente robusto para generar sincronización estable en redes grandes.

5. Conclusiones

A lo largo de este trabajo se ha hecho hincapié en la necesidad de obtener un entendimiento más profundo de los procesos biológicos que implican sincronización celular. Con este fin se ha propuesto un modelo para el reloj celular, una de las dos partes fundamentales que componen el reloj de segmentación, con el objetivo de estudiar la oscilación celular sincronizada que permite la somitogénesis. Los resultados obtenidos nos permiten concluir:

- En primer lugar, el modelo planteado para describir el reloj permite sincronizar de forma exitosa las oscilaciones de concentración de proteína Her/Hes en una población celular extensa.
- Los valores de los que depende esta sincronización son el Fold-Change de activación α y el *delay* τ presente en $D_{j\tau}$, que nos da información acerca de la escala temporal del mecanismo de interacción Delta-Notch. Obtenemos sincronización robusta ($R \approx 1$) para un amplio rango de valores de α y τ .
- El comportamiento cualitativo es siempre el mismo: Partiendo de una red desincronizada ($\alpha=1$) a medida que aumentamos α ganamos sincronización promedio R hasta alcanzar el estado sincronizado global ($R=1$). El sistema se mantiene sincronizado hasta alcanzar el α_{umbral} , valor a partir del cual la red se organiza en subgrupos con variaciones en amplitud y fase y se produce una desincronización gradual.
- El valor de α_{umbral} depende fuertemente de τ ; cuánto mayor es el *retraso* mayor es la región de α en la que se alcanza sincronización.
- El efecto de aumentar el tamaño de la red se traduce en una mayor dificultad en la sincronización global de esta.
- El efecto de incluir tanto ruido estático como estocástico en la simulación es similar. En ambos casos un aumento en el ruido se traduce en una mayor dispersión en las amplitudes y fases de la oscilación de cada célula, lo que hace que disminuya el valor de la sincronización promedio R para un α dado y en términos generales, que se dificulte el alcance de un estado sincronizado global. Un análisis algo más detallado de la influencia de ambos tipos de ruido en el sistema se desarrolla en el apéndice F.

Referencias

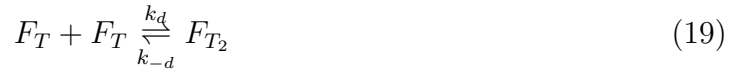
- [1] Beta, Carsten; Kruse, Karsten, 2017. Intracellular Oscillations and Waves, Annual Review of Condensed Matter Physics, Vol.8, Issue 1, pp.239-264.
- [2] Venzin, Olivier F.; Oates, Andrew C., 2019. What are you synching about? Emerging complexity of Notch signaling in the segmentation clock, Elsevier Science, Developmental Biology. En prensa.
- [3] Cooke, J., and Zeeman, E.C. (1976). A clock and wavefront model for control of the number of repeated structures during animal morphogenesis. *J. Theor. Biol.* 58, 455–476.
- [4] Hubaud, A., and Pourquié, O., 2014. Signalling dynamics in vertebrate segmentation. *Nat. Rev. Mol. Cell Biol.* 15, 709–721.
- [5] Oates, A. C.; Morelli, L. G.; Ares, S., 2012. Patterning embryos with oscillations: structure, function and dynamics of the vertebrate segmentation clock. *The Company of Biologists Ltd, Development (The Company of Biologists)*, Vol. 139, Issue 4, pp.625-639.
- [6] Daniela Roellig; Luis G. Morelli; Saúl Ares; Frank Jülicher; Andrew C. Oates, 2011. Enhanced SnapShot: The Segmentation Clock. Elsevier Science, Cell, Vol.145, Issue 5, pp.0-8000.
- [7] Horikawa, K., Ishimatsu, K., Yoshimoto, E., Kondo, S., Takeda, H., 2006. Noise-resistant and synchronized oscillation of the segmentation clock. *Nature* 441, 719–723.
- [8] Jiang, Y.-J., Aerne, B.L., Smithers, L., Haddon, C., Ish-Horowicz, D., Lewis, J., 2000. Notch signalling and the synchronization of the somite segmentation clock. *Nature* 408, 475.
- [9] Lewis, J., 2003. Autoinhibition with transcriptional delay: a simple mechanism for the zebrafish somitogenesis oscillator. *Curr. Biol.* 13, 1398–1408.
- [10] Morelli, L.G., Ares, S., Herrgen, L., Schröter, C., Jülicher, F., and Oates, A.C., 2009. Delayed coupling theory of vertebrate segmentation. *HFSP J.* 3, 55–66.
- [11] Yoshiki Kuramoto; Ikuko Nishikawa, 1987. Statistical macrodynamics of large dynamical systems. Case of a phase transition in oscillator communities, Springer, *Journal of Statistical Physics*, Vol.49, Issue 3-4, pp.569-605.
- [12] Morelli, Luis G.; Ares, Saúl; Herrgen, Leah; Schröter, Christian; Jülicher, Frank; Oates, Andrew C., 2009. Delayed coupling theory of vertebrate segmentation. Taylor and Francis Group, *HFSP Journal*, Vol.3, Issue 1, pp.55-66.

- [13] Rob Phillips, Jane Kondev, Julie Theriot, Hernan Garcia, 2012. Physical Biology of the Cell (2nd Edition), Garland Science, Chapter 3, pp.87-136.
- [14] Uri Alon, 2007, An introduction to systems biology: designs principles of biological circuits, Chapman & Hall/CRC, Chapter 2, pp.5-25.
- [15] Erin L. O'Brien; Elizabeth Van Itallie; Matthew R. Bennett, 2012. Modeling synthetic gene oscillators, Elsevier Science, Mathematical Biosciences, Vol.236, Issue 1, pp.0-15.
- [16] Formosa-Jordan P., Sprinzak D. (2014) Modeling Notch Signaling: A Practical Tutorial. In: Bellen H., Yamamoto S. (eds) Notch Signaling. Methods in Molecular Biology (Methods and Protocols), vol 1187. Humana Press, New York, NY.
- [17] Hasty, Jeff; Dolnik, Milos; Rottschäfer, Vivi; Collins, James J., 2002. Synthetic Gene Network for Entraining and Amplifying Cellular Oscillations, The American Physical Society, Physical Review Letters, Vol.88, Issue 14, pp.148101.
- [18] Garcia-Ojalvo, J.; Elowitz, M. B.; Strogatz, S. H., 2004. Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing, National Academy of Sciences, Proceedings of the National Academy of Sciences, Vol.101, Issue 30, pp.10955-10960.
- [19] Pablo Javier Blasco Hernández, 2016; Sistemas dinámicos y expresión de genes: cómo se forman los patrones celulares. Trabajo de Fin de Grado en Física, Universidad de Zaragoza, Facultad de Ciencias.

A. Funciones de Hill

Como ya se ha comentado, las funciones de Hill modelizan matemáticamente el término de producción presente en la ecuación diferencial que rige la dinámica de la concentración de una cierta proteína o especie intermedia. Esta producción será diferente dependiendo de si el gen que produce esta proteína está activado o reprimido. Veamos en detalle cuál es el procedimiento para llegar a la forma cualitativa de funciones de Hill reprimidas o activadas por un único factor de transcripción [15].

Para iniciar la activación o represión de un cierto gen, el factor de transcripción ha de unirse a una parte específica del promotor (operador). Dependiendo de su naturaleza esta unión aumentará o disminuirá el ritmo al que se une la ARNp al promotor y por tanto modificará el ritmo de producción de la proteína que codifica el gen. La tasa de producción de proteína dependerá por tanto de la cantidad de FT y de la probabilidad de que este se encuentre unido al operador. Consideremos el caso de un gen con un único promotor y un único operador y asumamos que el FT ha de dimerizarse previamente a su unión con el operador. Podemos escribir en este caso:



La ecuación (19) representa la dimerización del factor de transcripción: F_T indica la concentración de factor de transcripción, F_{T_2} la concentración de factor de transcripción dimerizado y k_d y k_{-d} las probabilidades por unidad de tiempo de dimerizar o romper la unión respectivamente. La ecuación (20) establece la reacción química en la que un FT dimerizado se une a una región del operador libre para dar lugar a un operador 'unido'. O_b y O_u representan la concentración de operador unido y sin unir. Si las reacciones están en equilibrio, entonces:

$$k_d F_T^2 = k_{-d} F_{T_2} \quad (21)$$

$$k_b O_u F_{T_2} = k_{-b} O_b \quad (22)$$

Además, se cumple que $O_u + O_b = N$, es decir, la concentración total de operador es constante. Con el sistema de ecuaciones que acabamos de plantear podemos despejar las concentraciones de operador unido O_b y de operador sin unir O_u en función del resto de parámetros:

$$O_u = \frac{N}{1 + \left[\frac{T_F}{K} \right]^2} \quad (23)$$

$$O_b = \frac{N \left[\frac{T_F}{K} \right]^2}{1 + \left[\frac{T_F}{K} \right]^2} \quad (24)$$

Las ecuaciones (23) y (24) son funciones de Hill.

Como la tasa de producción de una cierta proteína T_{Prod} en un circuito genético depende directamente de la concentración de operador al que se le ha unido el dímero de FT, podemos expresar el término de producción como:

$$T_{Prod} = \alpha_u O_u = \alpha_u \frac{N}{1 + \left[\frac{T_F}{K} \right]^2} \quad (25)$$

en el caso de una proteína expresada por un gen reprimido (por lo que sólo producirá proteínas en aquella fracción de operadores que no estén unidos a un dímero de FT) o como:

$$T_{Prod} = \alpha_b O_b = \alpha_b \frac{N \left[\frac{T_F}{K} \right]^2}{1 + \left[\frac{T_F}{K} \right]^2} \quad (26)$$

para una proteína expresada por un gen activado (por lo que sólo producirá proteínas en aquella fracción de operadores que estén unidos a un dímero de FT).

Cuando la unión FT-Operador requiere de una dimerización previa de FT el exponente del término $\frac{T_F}{K}$ es 2, pero esto no es así de forma general. El exponente n será equivalente al número de FT que tienen que polimerizar previamente a la unión con el operador. Este índice expresa por tanto la cooperatividad necesaria en la unión y se denomina exponente de Hill.

Así, las ecuaciones (25) y (26) generalizadas con un exponente de Hill arbitrario n representan el término de producción de la ecuación diferencial que rige la dinámica de la concentración de una proteína expresada por un gen reprimido o activado.

B. Delta-Notch 2 células: Ecuaciones y resultados

Se expone en este apéndice los pasos seguidos para dar lugar a las ecuaciones (6) y (7). Para un desarrollo más detallado se recomienda mirar en profundidad el tutorial de modelización de señalización Notch [16]. Escribiendo las ecuaciones del circuito genético en la figura 8 para la célula i se obtiene lo siguiente:

$$\frac{dD_i}{d\tau} = \frac{\alpha_d}{1 + \left(\frac{R_i}{\theta_r}\right)^h} - \gamma_d D_i \quad (27)$$

$$\frac{dR_i}{d\tau} = \frac{\alpha_r D_j^m}{\theta_d^m + D_j^m} - \gamma_r R_i \quad (28)$$

donde $\gamma_d, \gamma_r, \alpha_d, \alpha_r$ representan las tasas de degradación y tasas de máxima producción de ambas especies respectivamente, θ_d y θ_r hacen referencia a los umbrales de concentración de Delta y Notch necesarios para alcanzar la mitad de la represión o activación en la célula adyacente y m y h son los coeficientes de Hill de Notch y Delta respectivamente. En ambas ecuaciones el primer término es el de producción (ya sea activada o reprimida) y el segundo el término de degradación.

Sustituyendo i por j obtenemos las 2 ecuaciones diferenciales de la célula j .

Con el objetivo de reducir el gran número de parámetros presentes en las ecuaciones va a resultar conveniente adimensionalizarlas. Teniendo en cuenta que T_0, D_0, R_0 y t, d_i, r_i son las cantidades características dimensionales y adimensionales respectivamente de tiempo, ligando y represor, efectuamos los siguientes cambios de variable: $\tau = T_0 t$; $D_i = D_0 d_i$; $R_i = R_0 r_i$.

Además, escogiendo: $T_0 = \frac{1}{\gamma_r}$; $D_0 = \theta_d$; $R_0 = \theta_r$

Así:

$$\frac{D_0}{T_0} \frac{dd_i}{dt} = \frac{\alpha_d}{1 + \left(\frac{R_0 r_i}{R_0}\right)^h} - \gamma_d D_0 d_i \rightarrow \frac{dd_i}{dt} = \frac{\alpha_d}{\gamma_r \theta_d (1 + r_i^h)} - \frac{\gamma_d}{\gamma_r} d_i \quad (29)$$

$$\rightarrow \frac{dd_i}{dt} = \nu \left[\frac{\beta_d}{1 + r_i^h} - d_i \right] \quad (30)$$

donde hemos definido $\beta_d = \frac{\alpha_d}{\gamma_d \theta_d}$ y $\nu = \frac{\gamma_d}{\gamma_r}$ [16].

Procediendo de forma análoga para adimensionalizar la ecuación diferencial del represor obtenemos:

$$\frac{dr_i}{dt} = \frac{\beta_r d_j^m}{1 + d_j^m} - r_i \quad (31)$$

donde se ha usado $\beta_r = \frac{\alpha_r}{\gamma_r \theta_r}$.

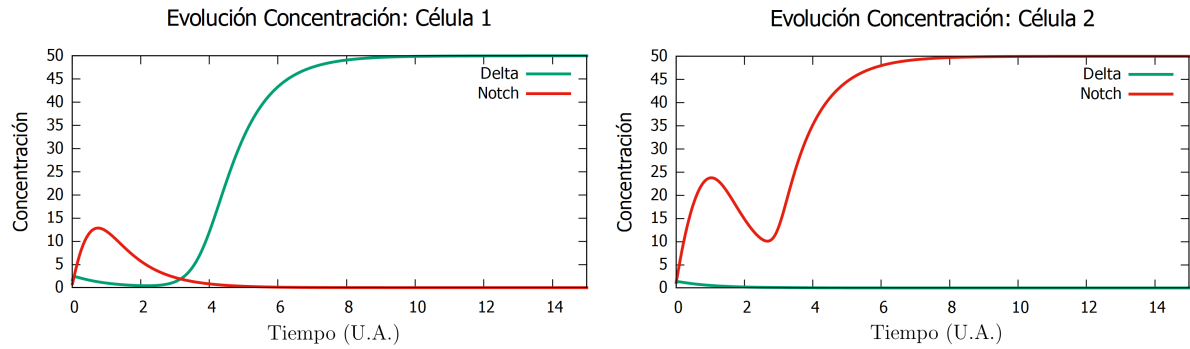


Figura 22: Evolución temporal de la concentración de las especies Delta y Notch de las 2 células que conforman el sistema

Resolviendo las ecuaciones para el sistema simple de dos células obtenemos dos soluciones. Cuando las concentraciones iniciales de Delta y de Notch son similares se llega a un equilibrio en el que las concentraciones de estas dos en cada célula es el mismo. Sin embargo, una pequeña diferencia en estas concentraciones da lugar a un estado de equilibrio distinto. Observando el circuito genético se observa que el Delta de la primera célula activa el represor Notch de la célula vecina y a su vez, este reprime el Delta de la célula vecina, lo que impide la proliferación del Delta de la primera célula. Por tanto, una pequeña variación en las condiciones iniciales desequilibra el sistema y da lugar a un proceso de amplificación de las diferencias en condiciones iniciales que tiende a un estado en el que una de las células está dominada por el ligando Delta mientras que su vecina está dominada por Notch, como se observa en la figura 22.

C. Métodos numéricos

A lo largo de las simulaciones se ha hecho uso de una gran cantidad de algoritmos; algunos de los más importantes son los siguientes:

- Como método de resolución numérica de las ecuaciones diferenciales que describen la dinámica de la concentración de la proteína Her/Hes en la red celular se ha hecho uso del conocido algoritmo de **Runge-Kutta de Orden 4**. El paso temporal h elegido es de $h = 0,01$
- Para asegurar la homogenización en los valores iniciales de ciertos parámetros como la concentración de proteína Her/Hes en el interior de cada una de las células de la red se necesita un algoritmo que genere números aleatorios sin correlaciones. Ha sido utilizado el algoritmo de **Parisi-Rapuano**
- Por último, para las simulaciones en las que se incluye ruido estocástico para simular una dinámica celular ruidosa (Apéndice F), se ha hecho uso del **algoritmo de Heun**, cuya particularidad es que la estocasticidad es representada en la dinámica mediante la inclusión de ruido multiplicativo en vez de aditivo. Este algoritmo calcula el valor de la concentración en un paso siguiente como:

$$x(t + \Delta t) = x + \frac{1}{2}[F(x) + F(\bar{x})]\Delta t + \frac{1}{2}[G(x) + G(\bar{x})]\Delta W \quad (32)$$

Este paso requiere el cálculo previo del denominado predictor, definido como:

$$\bar{x} = x + F(x)\Delta t + G(x)\Delta W \quad (33)$$

donde ΔW :

$$\Delta W = \sqrt{2C\Delta t}Z \quad (34)$$

con C el parámetro que cuantifica la magnitud del ruido térmico y Z un número aleatorio con distribución gaussiana generado a partir del **algoritmo de Box-Muller**. La inclusión de ruido estocástico modifica las ecuaciones diferenciales que describen la dinámica del sistema, que se dividen ahora en 2 sumandos: El primero de ellos se representa por $F(x)$ y su valor corresponde al que adquiriría la ecuación diferencial previamente a la adición de ruido, es decir, cuando integramos la dinámica del ligando Delta o de la proteína Her/Hes, $F(x)$ adquiere la forma de la función representada en las ecuaciones (13) o (14) respectivamente. El segundo sumando se describe como $G(x)$ y es la parte de la ecuación diferencial que multiplica al ruido $\xi(t)$ en las nuevas ecuaciones. Tomará el valor mostrado a continuación en caso de la resolución de la dinámica de Delta:

$$G(x) = \sqrt{\nu \left[\frac{\beta_d}{1 + H_i^h} + D_i \right]} \quad (35)$$

y el siguiente en caso de la resolución de la dinámica de la proteína Her/Hes:

$$G(x) = \sqrt{p_H \frac{1 + \alpha \langle D_{j\tau_2} \rangle^2}{[1 + \alpha \langle D_{j\tau_2} \rangle^2] [1 + H_{i\tau_1}^4]}} + \gamma_H H_i \quad (36)$$

D. Valores de parámetros usados en las simulaciones

Las ecuaciones diferenciales integradas en las simulaciones (Ecuaciones (13) y (14)) presentan una gran cantidad de parámetros cuyos valores han de ser fijados. Como ha sido comentado anteriormente los valores usados se caracterizan por generar oscilaciones desfasadas en ausencia de interacción celular ($\alpha=1$), es decir, son valores que permiten las oscilaciones individuales en la concentración de la proteína Her/Hes de cada célula. Para algunos parámetros se va a presentar sin embargo un rango de valores posibles en vez de un valor fijo. Esta dispersión respecto al valor principal representa la inclusión de ruido estático en ese parámetro como forma de modelizar las inhomogeneidades intrínsecas en los sistemas celulares extensos (Apéndice F).

- Paso temporal h : 0.01
- Cociente de tasas de degradación ligando-represor $\nu = \frac{\gamma_d}{\gamma_r} = 1$
- Producción de ligando $\beta_d = \frac{\alpha_d}{\gamma_d \theta_d} \in [29-31]$
- Producción proteína Her/Hes $p_H \in [0.95-1]$
- Tasa degradación Her/Hes $\gamma_H=0.1$

Además de los parámetros fijos, las ecuaciones cuentan con parámetros cuyo valor se modifica entre diferentes simulaciones. Para estos valores la notación ya no indica el efecto del ruido estático sino la región de valores de estos parámetros que ha sido explorada en las simulaciones:

- Fold-Change de activación $\alpha \in [1-3000]$.
La región de valores de α explorada en profundidad sólo abarca $\alpha \in [1-100]$
- *Delay* autorrepresión Her/Hes $\tau_1 \equiv \text{Delay}$ Notch $\tau_2 \equiv \tau = [600, 1200, 2000, 3500] \cdot h$ U.A.

Los *retrasos* y el tiempo en general están medidos en unidades de tiempo arbitrarias, es decir, τ no tiene unidades de segundos. Para poder establecer el tiempo real en segundos al que equivale una unidad arbitraria representada en las gráficas necesitamos conocer la relación entre el valor de la tasa de degradación propuesta en las simulaciones ($\gamma_H=0.1$) y su valor real. Conocida esa equivalencia podríamos calcular el periodo de las oscilaciones en las simulaciones (en segundos) y compararlo con las escalas temporales típicas del periodo de somitogénesis. Como no disponemos del dato de la tasa de degradación real de la proteína Her/Hes podemos plantearlo al revés: Dada la equivalencia entre U.A. y segundos podemos obtener una estimación de la tasa de degradación real y compararla con escalas típicas reales de ese parámetro.

La somitogénesis es un proceso biológico presente en una gran cantidad de especies, y cada especie presenta un periodo de oscilación característico diferente: En los peces cebra el periodo de las oscilaciones es del orden de 30 minutos mientras que en humanos se prolonga hasta las 4/5 horas. Nuestro propósito no es hacer una predicción precisa y exacta sino ver si los valores estimados de la tasa de degradación se acercan al orden de magnitud típico de γ , por lo que vamos a tomar como periodo de somitogénesis típico un valor intermedio entre los casos extremos propuestos, 120 minutos. Comparando este valor con el número de U.A. promedio que representan un periodo en nuestras simulaciones obtenemos la siguiente equivalencia:

$$1 \text{ U.A.} \approx 150 \text{ seg.} \quad (37)$$

Como además sabemos que la tasa de degradación propuesta en las simulaciones es $\gamma_H=0,1 \text{ U.A.}^{-1}$, la estimación del valor de γ_{real} queda:

$$\gamma_{real} \approx 2,5 \text{ h}^{-1} \quad (38)$$

Ya se ha comentado en párrafos anteriores que no tenemos un valor de la tasa de degradación de la proteína Her/Hes para comparar con la estimación, pero γ_{real} se encuentra dentro del rango de valores típicos de tasa de degradación para una proteína.

E. Valores de α_{umbral} en función de τ y tamaño de red

Como hemos discutido en las secciones anteriores, el comportamiento cualitativo de R vs α es el mismo para diferentes valores de *delay* τ , pero el α_{umbral} a partir del que la sincronización en la red deja de ser la unidad difiere enormemente entre diferentes valores de *delay*, como se muestra en las siguientes tablas:

Red 6x6	$\tau = 6$ (U.A.)	$\tau = 12$ (U.A.)	$\tau = 20$ (U.A.)	$\tau = 35$ (U.A.)
α_{umbral}	10	20	≈ 800	$(1-4) \cdot 10^5$

Para la red 12x12 y *delay* $\tau=6$ no se llega a alcanzar la sincronización completa ($R=1$) por lo que el parámetro α_{umbral} en este caso carece de sentido. Sucede lo mismo para la red 18x18 y $\tau=6$ U.A.

Red 12x12	$\tau = 6$ (U.A.)	$\tau = 12$ (U.A.)	$\tau = 20$ (U.A.)
α_{umbral}	-	20	≈ 500

Red 18x18	$\tau = 6$ (U.A.)	$\tau = 12$ (U.A.)
α_{umbral}	-	≈ 20

F. Ruido estático y estocástico

Cuando hablamos de introducir ruido en las ecuaciones podemos hacerlo de dos formas:

- Ruido estático: A pesar de que las redes celulares que se plantean en el modelo están compuestas únicamente por un tipo de célula, esto no significa que estas sean copias idénticas unas de otras; de hecho, a pesar de su aparente homogeneidad existen pequeñas diferencias composicionales que las hacen únicas. Es conveniente por tanto modelizar estas pequeñas inhomogeneidades intrínsecas a la red introduciendo en las simulaciones mínimas desviaciones aleatorias respecto al valor fijado de ciertos parámetros.
- Ruido estocástico: Los sistemas biológicos son sistemas ruidosos por naturaleza debido a la estocasticidad presente en las cinéticas celulares. El procedimiento para introducir este tipo de ruido en las simulaciones es más sofisticado; para evaluar las diferencias en el alcance de sincronización cuando se contempla o no ruido estocástico se ha modelizado este ruido en la simulación mediante el algoritmo de Heun.

En los resultados presentados en la sección (4) del trabajo las simulaciones incluían ruido estático moderado (Apéndice D) y no incluían ruido estocástico. Este apéndice trata de evaluar como la obtención de sincronización se ve modificada por valores mayores de ruido estático y por la introducción de ruido estocástico.

F.1. Ruido estático

En esta simulación comparativa se han utilizado rangos de valores extensos (ruido estático alto) en dos parámetros de las ecuaciones: Cada célula tendrá de forma aleatoria un valor de β_d entre 20 y 40 y un valor p_H que oscila entre 0.6 y 1. Como observamos, en esta simulación se incrementa considerablemente la magnitud del ruido estático respecto a la utilizada en las simulaciones con las que se han obtenido los resultados de la sección (4). La idea de este aumento radica en que en este caso estamos interesados únicamente en discernir qué diferencias induce en la red y en la sincronización la presencia de ruido estático, por lo que cuanto mayor sea su magnitud mejor idea nos podremos hacer de como afecta al sistema.

En la figura 23 se muestra la evolución de la concentración de las 36 células de una red 6x6 para un valor de α para el que con valores más moderados de ruido estático se conseguía sincronización completa.

Como podemos comprobar, el efecto de aumentar el ruido estático no destruye la sincronización pero aumenta la dispersión tanto en amplitudes de oscilación como en las fases. Esta pequeña dispersión en las fases provoca que el valor promedio de sincronización R descienda respecto a la simulación sin ruido, como podemos apreciar en la figura 24.

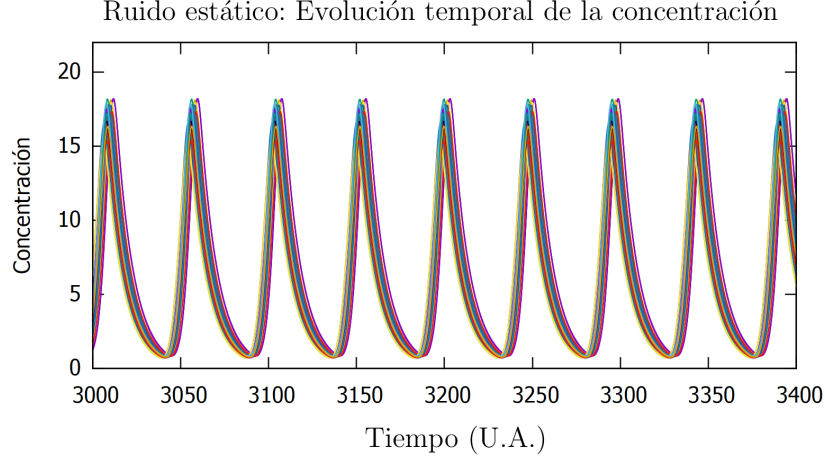


Figura 23: Red 6x6: Concentración en función del tiempo de la proteína Her/Hes de las 36 células que componen la red. Se ha usado un valor de $\tau=12$ U.A. y un $\alpha=6$.

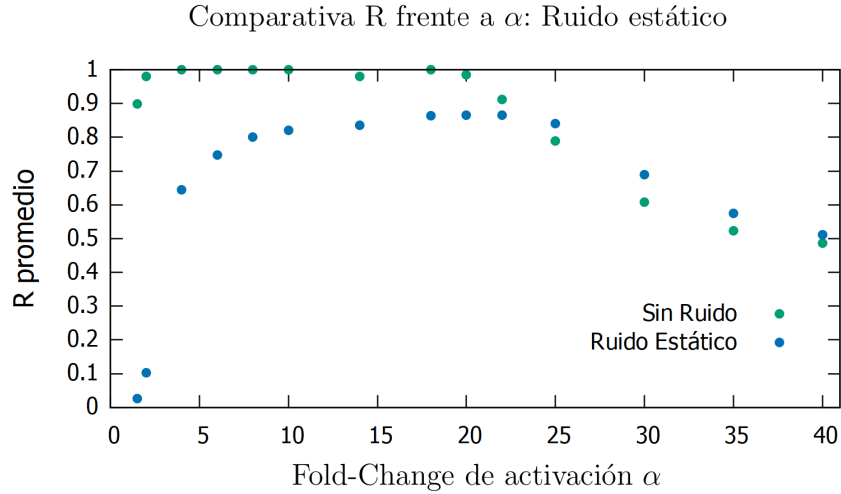


Figura 24: Red 12x12: Comparativa de la evolución de la sincronización R frente a α con y sin introducción de ruido estático en las ecuaciones. Se ha usado un *delay* $\tau=12$ U.A.

F.2. Ruido estocástico

Como se detallaba en el apéndice C, al integrar las ecuaciones diferenciales de la dinámica resultantes tras la introducción de ruido multiplicativo en el sistema, el parámetro que cuantifica el nivel de ruido estocástico es C . Para evaluar cómo varía el alcance de sincronización en el sistema celular cuando consideramos este tipo de ruido es conveniente comparar la evolución de la sincronización promedio frente a α para diferentes valores de nivel de ruido C , como se detalla en la figura 25

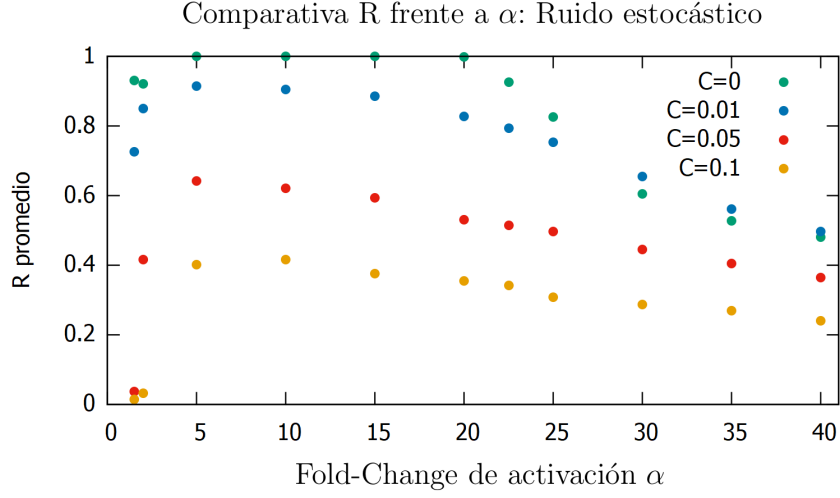


Figura 25: Red 12x12: Comparativa de la evolución de la sincronización R frente a α para diferentes valores de nivel de ruido estocástico C . Se ha usado un *delay* $\tau=12$ U.A.

Cuando $C=0$ no se introduce ruido en el sistema y se recuperan los resultados obtenidos en el caso "Sin Ruido" de la gráfica 24. Como cabía esperar, a medida que aumentamos el nivel de ruido el sistema exhibe mayores dificultades para mantener las oscilaciones de diferentes células en fase, aumenta la dispersión y disminuye la sincronización promedio, es decir, se confirma el hecho intuitivo de que la introducción de ruido estocástico (al igual que sucedía con el estático) dificulta las dinámicas que llevan a la sincronización de la red celular.

G. Códigos de las simulaciones

Para la obtención de los resultados expuestos se han desarrollado una buena cantidad de programas en el lenguaje C. Muchos de ellos presentan partes de código comunes, por lo que en esta sección no se va a presentar cada código de forma individual sino que se va a mostrar de forma detallada el código más completo desarrollado indicando sus subalgoritmos más relevantes. Generalmente para la obtención de resultados no se ha utilizado este código en su forma completa, sino que se han desarrollado otros programas que obtienen de forma más eficiente y específica el resultado requerido. Además, en el código presentado se resuelven las ecuaciones diferenciales definitivas (16) y (17). Como es lógico, para la obtención de resultados previos a los presentados en la sección 4 las ecuaciones diferenciales resueltas son distintas.

G.1. Programa principal

El programa principal calcula la evolución temporal de la concentración de la proteína Her/Hes de cada una de las células que componen la red celular extensa. Tiene la opción de registrar las concentraciones de todas las especies en el interior celular de cada una de las células de la red, de calcular el parámetro de orden en cada instante de tiempo y de generar a la vez que se ejecuta el programa un vídeo en tiempo real en el que se observa la evolución temporal de la concentración de Her/Hes de cada célula de forma visual. De este tipo de videos se han obtenido por ejemplo, figuras como la 18.

G.1.1. Cabecera y definiciones

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<math.h>
4 #include<time.h>
5
6 #define V 1
7 //Cociente de las tasas de degradacion ligando-represor
8 #define BetaD 30
9 //Parametro relacionado con la produccion del ligando
10 #define TTB 0.95
11 //Parametro relacionado con la produccion de la proteina Her/Hes
12 #define Alfa 12
13 //Fold Change de la activacion de Her/Hes
14 #define Gammah 0.1
15 //Coeficiente de degradacion de la proteina Her/Hes
16 #define p 4
```

```

17 //Coef. Hill del represor
18 #define h 0.01
19 //Paso Temporal
20 #define Tiempo 800000
21 //Numero de pasos de tiempo h realizados
22 #define Lx 12
23 //Numero de celulas en la coordenada x
24 #define Ly 12
25 //Numero de celulas en la coordenada y
26 #define N 144
27 //Numero total de celulas (N=Lx*Ly)
28 #define a 1
29 //Lado del hexagono (Cada celula se representa por un hexagono)
30 #define tau 1200
31 //Desfase
32 #define Terma 400000
33 //Tiempo de termalizacion
34 #define NormRANu (2.3283063671E-10F
35 //Constante usada en
36
37 FILE*fLig;
38 FILE*fRep;
39 FILE*gnuplotPipe;
40 FILE*fCoord1NR;
41 FILE*fPOrden;
42 FILE*fR;

```

G.1.2. Algoritmo de Parisi-Rapuno

```

1  \\Generador de numeros aleatorios entre 0 y 1 sin correlaciones
2
3  unsigned int irr[256];
4  unsigned int ir1;
5  unsigned char ind_ran,ig1,ig2,ig3;
6  extern float Random(void);
7  extern void ini_ran(int SEMILLA);
8
9  float Random(void)
10 {
11  float r;
12  ig1=ind_ran-24;
13  ig2=ind_ran-55;

```

```

14  ig3=ind_ran-61;
15  irr[ind_ran]=irr[ig1]+irr[ig2];
16  ir1=(irr[ind_ran]^irr[ig3]);
17  ind_ran++;
18  r=ir1*NormRANu;
19
20  return r;
21  }
22  void ini_ran(int SEMILLA)
23  {
24  int INI,FACTOR,SUM,i;
25  srand(SEMILLA);
26  INI=SEMILLA;
27  FACTOR=67397;
28  SUM=7364893;
29  for(i=0; i<256; i++)
30  {
31  INI=(INI*FACTOR+SUM);
32  irr[i]=INI;
33  }
34  ind_ran=ig1=ig2=ig3=0;
35  }

```

G.1.3. Runge-Kutta de Orden-4

```

1  //Nucleo del algoritmo utilizado para resolver de forma
2  //aproximada ecuaciones diferenciales.
3
4  double FuncionLig(double rep, double lig,double*PBeta,int i)
5  //Subalgoritmo de evolucion del ligando Delta
6  {
7  double Lpunto;
8  Lpunto= V*((PBeta[i]/(1+pow(rep,p)))-lig;
9
10 return Lpunto;
11 }
12 double FuncionH(double rep, double repold,
13                 double ligm, double*TasaTB,int i,double*Alpha)
14 //Subalgoritmo de evolucion de la proteina Her/Hes
15 {
16 double Hpunto;
17 Hpunto= TasaTB[i]*((1+(Alpha[i]*(ligm*ligm)))/

```

```

18  ((1+(ligm*ligm))*(1+(repold*repold*repold*repold))))-Gammah*rep;
19  return Hpunto;
20  }
21
22  double Runge-KuttaLigando(double rep, double lig, double *PBeta, int i)
23  //Subalgoritmo de evolucion del ligando delta por RK
24  {
25  double k1=0, k2=0, k3=0, k4=0;
26  double aux2, aux3, aux4;
27  double Ligf;
28  Ligf=0;
29  k1=FuncionLig(rep, lig, PBeta, i);
30  aux2=(lig+(0.5*k1*h));
31  k2=FuncionLig(rep, aux2, PBeta, i);
32  aux3=(lig+(0.5*k2*h));
33  k3=FuncionLig(rep, aux3, PBeta, i);
34  aux4=lig+k3*h;
35  k4=FuncionLig(rep, aux4, PBeta, i);
36  Ligf=lig+(h*((k1+2*k2+2*k3+k4)/6));
37  return Ligf;
38  }
39
40
41  double Runge-KuttaRepresor(double rep, double repold,
42                             double ligm, double *TasaTB, int i, double*Alpha)
43  //Subalgoritmo de evolucion de Her/Hes por RK
44  {
45  double k1, k2, k3, k4;
46  double aux2, aux3, aux4;
47  double Repf;
48  Repf=0;
49  k1=FuncionH(rep, repold, ligm, TasaTB, i, Alpha);
50  aux2=(rep+(0.5*k1*h));
51  k2=FuncionH(aux2, repold, ligm, TasaTB, i, Alpha);
52  aux3=(rep+(0.5*k2*h));
53  k3=FuncionH(aux3, repold, ligm, TasaTB, i, Alpha);
54  aux4=rep+k3*h;
55  k4=FuncionH(aux4, repold, ligm, TasaTB, i, Alpha);
56
57  Repf=rep+ (h*((k1+2*k2+2*k3+k4)/6));
58  return Repf;
59
60  }

```

G.1.4. Algoritmo que devuelve posiciones de los vecinos

Este algoritmo está diseñado para devolver los índices que codifican la posición de los 6 vecinos de una cierta célula i . Para entender como funciona este subalgoritmo es preciso visualizar previamente cuál es la indexación de la red celular. Esta indexación es prácticamente equivalente a la mostrada en la figura 26, con la diferencia de que las columnas pares son más bajas que las impares, es decir, si la red de la figura 26 subiera cada célula de sus columnas impares una posición, se llegaría a la indexación planteada en este trabajo. Las condiciones que nos dirán si una cierta célula indexada i pertenece o no a una cierta región de la red celular son largas y sutiles ya que estas dependen críticamente del tamaño de la red. El algoritmo tiene en cuenta estas sutilezas y será válido para encontrar independientemente del tamaño de la red y de la posición de la célula en la red los índices que indexan los 6 vecinos de una cierta célula i siempre y cuando la dimensión de la red sea la misma en ambas direcciones y además sea par (6x6, 8x8, 12x12, 16x16, etc).

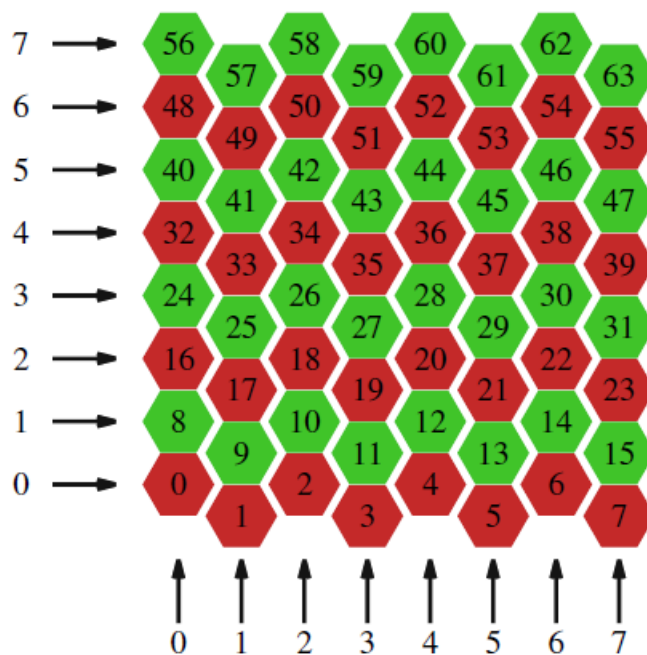


Figura 26: Criterio utilizado para indexar la red celular hexagonal. En este ejemplo se indexa mediante este criterio una red de dimensión 8x8 [19].

```

1 //Este subalgoritmo recibe un indice i que indica la posicion de una
2 //cierta celula en la red celular y devuelve independientemente de la
3 //posicion de esta (interior de la red, extremo, esquina, etc) y del
4 //tamano de la red, los indices de las posiciones de sus 6 celulas vecinas.
5 //Recordamos que sea cual sea la posicion de la celula en la red, esta

```

```

6 //contara con 6 vecinos ya que se esta haciendo uso de condiciones de contorno
7 // periodicas.
8
9 void CondContorno(int*v1, int*v2, int*v3, int*v4, int*v5, int*v6, int i)
10 {
11     int lx, ly;
12     ly= (int)(i/Lx);
13     lx= i-(ly*Lx);
14
15     if (lx>0 && lx<(Lx-1) && ly>0 && ly<(Ly-1)&& Resto(lx/2)==0)
16         //Condicion y vecinos para el caso de celula en una columna interior par
17         {
18             *v1= i+1;
19             *v2= i-1;
20             *v3= i+Lx;
21             *v4= i-Lx;
22             *v5=i-(Lx+1);
23             *v6=i-(Lx-1);
24         }
25     else if (lx>0 && lx<(Lx-1) && ly>0 && ly<(Ly-1)&& Resto(lx/2)!=0)
26         //Condicion y vecinos para el caso de celula en una columna interior impar
27         {
28             *v1= i+1;
29             *v2= i-1;
30             *v3= i+Lx;
31             *v4= i-Lx;
32             *v5=i+(Lx+1);
33             *v6=i+(Lx-1);
34         }
35     else if ((lx==0 && i!=0 && i!=(N-Lx)) ||
36              (lx==(Lx-1) && i!=(Lx-1) && i!=N-1))
37         //Condicion y vecinos para el caso de celula en el borde dcho. o izq.
38         //de la red (Exceptuando esquinas)
39         {
40             *v1= i+1;
41             *v2= i-1;
42             *v3= i+Lx;
43             *v4= i-Lx;
44             *v5=i-(Lx-1);
45             *v6=i+(Lx-1);
46         }

```

```

47
48 else if ((ly==0 && i!=0 && i!=Lx-1 && Resto(lx/2)==0))
    //Condicion y vecinos para el caso de celula en el borde inferior
49 // columna par (Exceptuando esquinas)
50 {
51     *v1= i+1;
52     *v2= i-1;
53     *v3= i+Lx;
54     *v4= i+(N-Lx);
55     *v5=i+(N-Lx+1);
56     *v6=i+(N-Lx-1);
57 }
58
59 else if ((ly==0 && i!=0 && i!=Lx-1 && Resto(lx/2)!=0))
    //Condicion y vecinos para el caso de celula en el borde inferior
60 // columna impar (Exceptuando esquinas)
61 {
62     *v1= i+1;
63     *v2= i-1;
64     *v3= i+Lx;
65     *v4= i+(N-Lx);
66     *v5=i+(Lx+1);
67     *v6=i+(Lx-1);
68 }
69
70 else if ((ly==(Ly-1) && i!=(N-Ly) && i!=(N-1) && Resto(lx/2)==0))
    //Condicion y vecinos para el caso de celula en el borde superior
71 // columna par (Exceptuando esquinas)
72 {
73     *v1= i+1;
74     *v2= i-1;
75     *v3= i-Lx;
76     *v4= i-(N-Lx);
77     *v5=i-(Lx+1);
78     *v6=i-(Lx-1);
79 }
80
81 else if ((ly==(Ly-1) && i!=(N-Ly) && i!=(N-1) && Resto(lx/2)!=0))
    //Condicion y vecinos para el caso de celula en el borde superior
82 // columna impar (Exceptuando esquinas)
83 {
84     *v1= i+1;
85     *v2= i-1;

```

```

86  *v3= i-Lx;
87  *v4= i-(N-Lx);
88  *v5=i-(N-Lx+1);
89  *v6=i-(N-Lx-1);
90  }
91
92  else if (i==0)
    //Condicion y vecinos para el caso de celula en la esquina inferior izq.
93  {
94  *v1= i+1;
95  *v2= i+(N-1);
96  *v3= i+Lx;
97  *v4= i+(N-Lx);
98  *v5=i+(Lx-1);
99  *v6=i+(N-Lx+1);
100 }
101
102 else if (i==(Lx-1))
    //Condicion y vecinos para el caso de celula en la esquina inferior dcha.
103 {
104 *v1= i+1;
105 *v2= i-1;
106 *v3= i+Lx;
107 *v4= i+(Lx-1);
108 *v5=i-(Lx-1);
109 *v6=i+(N-Lx);
110 }
111
112 else if (i==(N-Lx))
    //Condicion y vecinos para el caso de celula en la esquina superior izq.
113 {
114 *v1= i+1;
115 *v2= i-1;
116 *v3= i-Lx;
117 *v4= i-(Lx-1);
118 *v5=i+(Lx-1);
119 *v6=i-(N-Lx);
120 }
121
122 else if (i==(N-1))
    // //Condicion y vecinos para el caso de celula en la esquina superior dcha.
123 {
124 *v1= i-1;

```



```

125 *v2= i-(N-Lx);
126 *v3= i-Lx;
127 *v4= i-(N-1);
128 *v5=i-(Lx-1);
129 *v6=i-(N-Lx+1);
130 }
131
132 }

```

G.1.5. Algoritmos para representación gráfica en GNUPlot

```

1 //El primer subalgoritmo da las coordenadas para la representacion
2 // espacial en GNUPlot de la concentracion de proteina Her/Hes
3 // de cada celula de la red a partir del indice que nos indica su
4 // posicion en la red.
5 void CoordHex(double CRepresorV[][tau+1], FILE*fCoord, int k)
6 {
7     int i,j;
8     for(i=0; i<Ly; i++)
9     {
10         for(j=0; j<Lx; j++)
11         {
12             if(Resto(j/2)==0) //Condicion para columna par
13             {
14                 fprintf(fCoord, "f f f\n",0.5+(j*0.5*a),0.5+(i*sqrt(3)*a)
15                                     ,CRepresorV[(i*Ly)+j][k]);
16             }
17             else
18             {
19                 fprintf(fCoord, "f f f\n",0.5+(0.5*a*j),0.5+((sqrt(3)/2)*a)+i*sqrt(3)*a
20                                     ,CRepresorV[(i*Ly)+j][k]);
21             }
22         }
23     }
24 }
25
26 //El segundo subalgoritmo abre GNUPlot mientras se ejecuta el programa
27 // y pinta cada celula en la posicion de la red que le corresponde,
28 // dando informacion de la concentracion de Her/Hes en su interior
29 // mediante un codigo de colores.
30 void PintaGNUPlot (void)
31 {

```

```

32 fprintf(gnuplotPipe,"plot 'Coord1NRDD.txt'
33          u 1:2:3 w points pt 7 ps 5 lc palette\n");
34 }

```

G.1.6. Main del programa

```

1  int main()
2  {
3  //Declaracion de variables
4  int i,j,k;
5  double TasaTB[N];
6  double PBeta[N];
7  double Alpha[N];
8  double CLigandoV[N][tau+1]; //Guarda el valor del ligando Delta para cada
9  //celula en un cierto instante y en tau instantes anteriores.
10 double CRepresorV[N][tau+1]; //Guarda el valor de Her/Hes para cada
11 //celula en un cierto instante y en tau instantes anteriores.
12 double VectRep[N], VectRepC[N], Resta[N]; //Vectores usados en el calculo
13 //del parametro de orden R.
14 double MF=0,MAux=0,MC=0; //Variables usadas en el calculo de R.
15 double R=0; //Parametro de orden normalizado
16 ini_ran(time(NULL)); //Semilla para el generador de numeros aleatorios.
17 //Apertura de ficheros
18 fLig=fopen("LigNRDD.txt","w");
19 fRep=fopen("RepNRDD.txt","w");
20 fPOrden=fopen("POrdenNRDD.txt","w");
21 fR=fopen("POrdenR.txt","w");
22 //Condiciones iniciales para todos los parametros.
23 //Se muestra un caso particular arbitrario. Podria ponerse otro rango
24 // de variacion de ruido estatico o incluso no ponerse; de hecho, los
25 // resultados obtenidos no presentaban ruido estatico en el parametro Alpha.
26 //
27 for (k=0; k<N; k++)
28 {
29 TasaTB[k]=TTB +(double)(Random()/20);
30 //Ruido estatico: TTB varia entre 0.95 y 1 en las celulas.
31 PBeta[k]=BetaD +(double)(((Random()-0.5))*2);
32 //Ruido estatico: BetaD varia entre 29 y 31 en las celulas.
33 Alpha[k]=Alfa +(double)(((Random()-0.5))*0.1);
34 //Ruido estatico: Alpha varia entre 1.70 y 1.80 en las celulas.
35 //Inicializamos a 0 todas las componentes de los vectores principales
36 //del programa.

```

```

37 VectRep[k]=0;
38 VectRepC[k]=0;
39 Resta[k]=0;
40 }
41 //Conjunto de comandos para comunicar el programa en C con el programa
42 //de representacion grafica GNUPlot
43 gnuplotPipe= popen("C:\\gnuplot\\bin\\gnuplot.exe","w");
44 //Abrimos GNUPlot
45 fprintf(gnuplotPipe,"set cbrange [1.0:11.5]\\n");
46 //Ajustamos el rango de variacion de la coordenada z, que en nuestro caso
47 //es mostrada mediante un codigo de colores y representa la concentracion
48 //de proteina Her/Hes.
49 fprintf(gnuplotPipe,"set xrange [0:6.5]\\n");
50 //Ajustamos el rango en el eje x
51 fprintf(gnuplotPipe,"unset key\\n");
52 //Eliminamos la leyenda
53 fprintf(gnuplotPipe,"set title font ',14'\\n");
54 //Ajustamos la fuente del titulo
55 fprintf(gnuplotPipe,"set xlabel 'Coordenada espacial x'\\n");
56 fprintf(gnuplotPipe,"set ylabel 'Coordenada espacial y'\\n");
57 fprintf(gnuplotPipe,"set title 'Concentracion Her/Hes celular: Red 12x12'\\n");
58 //Ponemos el titulo de la grafica y de los ejes.
59 fprintf(gnuplotPipe,"cd 'C:\\Users\\raul\\OneDrive\\
60                                     Escritorio\\TFG\\Programa'\\n");
61 //Ponemos el directorio en el que se guarda el archivo a representar.
62 fflush(gnuplotPipe);
63
64 for (i=0; i<N; i++){
65 //Elegimos las condiciones iniciales para la primera componente de la
66 // dimension "tiempo" de los vectores que guardan las concentraciones
67 //de las especies.
68     if(Resto(i/2)==0) //Se da la opcion de dar condiciones iniciales
69     //distintas en funcion de si la columna es par o impar.
70     {
71         CLigandoV[i][0]=68*Random();
72         CRepresorV[i][0]=25*Random();
73     }
74     else if(Resto(i/2)!=0)
75     {
76         CLigandoV[i][0]=4*Random();
77         CRepresorV[i][0]=58*Random();
78     }
79 }

```

```

80  for(i=0; i<N; i++){
81  //Inicializamos a cero el resto de componentes temporales.
82      for(j=1; j<tau+1; j++)
83          {
84              CLigandoV[i][j]=0;
85              CRepresorV[i][j]=0;
86          }
87  }
88
89  for (j=1; j<Tiempo; j++) //Bucle temporal
90  {
91      fCoord1NR=fopen("Coord1NRDD.txt","w");
92      //Hay ligeras diferencias en los comandos en funcion de si el contador
93      //temporal es mayor o menor que tau+1, por lo que se anade un condicional.
94      if(j<tau+1) //Caso de tiempo menor que tau+1
95      {
96          for (i=0; i<N; i++) //Bucle a lo largo del numero de celulas
97          {
98              int v1=0,v2=0,v3=0,v4=0,v5=0,v6=0;
99              double Ligm;
100              CondContorno(&v1,&v2,&v3,&v4,&v5,&v6,i); //Calculo de vecinos
101              Ligm=((CLigandoV[v1][0]+CLigandoV[v2][0]+CLigandoV[v3][0]
102                  +CLigandoV[v4][0]+CLigandoV[v5][0]+CLigandoV[v6][0])/6);
103              //Calculo de la concentracion de ligando promedio de las celulas vecinas.
104              //Cuando el paso temporal es menor al delay, el sistema aun no
105              //puede percibir el efecto del delay y no se puede usar un valor
106              // un tiempo tau anterior porque aun no ha sido generado.
107              //Mientras se cumpla la condicion del bucle, la concentracion
108              //un tiempo tau anterior tendra que ser la concentracion inicial.
109
110              CLigandoV[i][j]=RungeKuttaLigando(CRepresorV[i][j-1]
111                  ,CLigandoV[i][j-1],PBeta,i);
112              //Calculo de concentracion Delta un tiempo h posterior
113              CRepresorV[i][j]=RungeKuttaRepresor(CRepresorV[i][j-1]
114                  ,CRepresorV[i][0],Ligm,TasaTB,i,Alpha);
115              //Calculo de concentracion Her/Hes un tiempo h posterior
116          }
117      }
118      else if (j>=tau+1) //Caso en que el tiempo es mayor que tau+1
119      {
120          for(i=0; i<N; i++)
121          {
122              for(k=0; k<tau; k++)

```

```

123 //Los vectores tienen tamano tau+1, por lo que cuando el tiempo
124 //es mayor que tau tenemos que eliminar el valor de concentracion
125 //mas viejo para hacer hueco para el nuevo valor
126 //
127 {
128     CLigandoV[i][k]=CLigandoV[i][k+1];
129     CRepresorV[i][k]=CRepresorV[i][k+1];
130 }
131 }
132
133 for (i=0; i<N; i++)
134 {
135     int v1=0,v2=0,v3=0,v4=0,v5=0,v6=0;
136     double Ligm;
137     CondContorno(&v1,&v2,&v3,&v4,&v5,&v6,i);
138     Ligm=((CLigandoV[v1][0]+CLigandoV[v2][0]+CLigandoV[v3][0]
139           +CLigandoV[v4][0]+CLigandoV[v5][0]+CLigandoV[v6][0])/6);
140 //La componente 0 del vector siempre guarda la concentracion en tau
141 // pasos atras, por lo que cuando se pasa la primera componente del vector
142 //se esta pasando la concentracion en un tiempo t-tau.
143 //De esta forma se modeliza computacionalmente los desfases tau propuestos
144 //en las ecuaciones diferenciales.
145 CLigandoV[i][tau]=Runge_KutaLigando(CRepresorV[i][tau-1]
146                                     ,CLigandoV[i][tau-1],PBeta,i);
147 CRepresorV[i][tau]=Runge_KutaRepresor(CRepresorV[i][tau-1]
148                                     ,CRepresorV[i][0],Ligm,TasaTB,i,Alpha);
149 }
150 }
151 //Se escribe en los ficheros la concentracon de ligando Delta
152 //y de proteina Her/Hes para cada paso de tiempo.
153 for(i=0; i<N; i++)
154 {
155     if(i==0)
156     {
157         fprintf(fLig,"f",j*h;
158         fprintf(fRep,"f",j*h;
159     }
160     if(j<tau+1)
161     {
162         fprintf(fLig,"f",CLigandoV[i][j];
163         fprintf(fRep,"f",CRepresorV[i][j];
164     }
165     else

```

```

166     {
167         fprintf(fLig,"f",CLigandoV[i][tau];
168         fprintf(fRep,"f",CRepresorV[i][tau];
169     }
170 }
171 fprintf(fLig,"\n");
172 fprintf(fRep,"\n");
173 //Calculo de variables necesarias para la obtencion del parametro
174 // de orden (Termalizamos previamente)
175 if(j>Terma){
176     double M=0;
177     for(i=0;i<N;i++){
178         M=M+(CRepresorV[i][tau]/N);
179         VectRep[i]=VectRep[i]+CRepresorV[i][tau];
180         VectRepC[i]=VectRepC[i]+(CRepresorV[i][tau]*CRepresorV[i][tau]);
181     }
182     MAux=MAux+M;
183     MC=MC+(M*M);
184 }
185 //Condicion para poder sacar gifs de tiempos anteriores al valor
186 //del desfase.
187 if(j<tau+1)
188 {
189     k=j;
190 }
191 else
192 {
193     k=tau;
194 }
195 CoordHex(CRepresorV,fCoord1NR,k); //Llamamos al algoritmo que
196 //calcula las coordenadas en las que va cada celula para la
197 //representacion espacial de las concentraciones.
198 if(j>200000){ //Esta condicion controla el momento en que empieza
199 //a ejecutarse el gif
200     if(Resto(j/50)==0){ //Para no saturar el programa de representacion
201 //se da la opcion de que solo se pinte cada ciertos pasos temporales.
202         PintaGNUPlot(); //Se llama al programa que da lugar a la
203 //representacion espacial de las concentraciones.
204     }
205 }
206 fclose(fCoord1NR);
207 }
208

```

```

209 MC=(double)(MC/(Tiempo-Terma));
210 //Valor medio de M(t)*M(t)
211 MF=(double)(MAux/(Tiempo-Terma));
212 //Valor medio de M(t)
213 MF=(MF*MF);
214 //Valor medio de M(t) al cuadrado
215
216 //Calculamos el denominador del parametro de orden
217 for (i=0;i<N;i++)
218 {
219 VectRep[i]=(double)(VectRep[i]/(Tiempo-Terma));
220 //Valor medio de b_i(t)
221 VectRepC[i]=(double)(VectRepC[i]/(Tiempo-Terma));
222 //Valor medio de b_i(t)*b_i(t)
223 VectRep[i]=(VectRep[i]*VectRep[i]);
224 //Valor medio de b_i(t) elevado al cuadrado
225 Resta[i]=VectRepC[i]-VectRep[i];
226 }
227 double Den=0; //Denominador del parametro de orden R
228 for (i=0;i<N;i++)
229 {
230 //Promedio sobre celulas
231 Den=Den+Resta[i];
232 }
233
234 Den=(double)(Den/N); //Denominador de R
235 R=(double)((MCMF)/Den); //Parametro de orden R
236 fprintf(fR,"f \n",R);
237
238 //Cerramos todos los ficheros que han sido abiertos
239
240 fclose(fLig);
241 fclose(fRep);
242 pclose(gnuplotPipe);
243 fclose(fPOrden);
244 fclose(fR);
245 }

```