



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

Modelo computacional basado en agentes de un reloj
multicelular sintético acoplado por percepción de
quórum

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Biólogo

P R E S E N T A:

Paul Damian Jiménez Nuño

DIRECTOR DE TESIS:
Dr. Pablo Padilla Longoria

2016

0 Ciudad Universitaria, CDMX





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno
Jiménez
Nuño
Paul Damian
46236355
Universidad Nacional Autónoma de
México
Facultad de Ciencias
Biología
306261604
2. Datos del Tutor
Dr
Pablo
Padilla
Longoria
3. Datos del sinodal 1
Dra
Carolina
Barriga
Montoya
4. Datos del sinodal 2
M.en C
Miguel
Lara
Aparicio
5. Datos del sinodal 3
M.en C
José Agustín
Mercado
Reyes
6. Datos del sinodal 4
Biól
Eniak
Hernández
Alarcón
7. Datos del trabajo escrito
Modelo computacional basado en agentes de un reloj multicelular sintético acoplado
por percepción de quórum
88 p
2016

Modelo computacional basado en agentes
de un reloj multicelular sintético acoplado
por percepción de quórum

Agradecimientos

A mis padres Edi y Martin por todo su apoyo y cariño que me han brindado a lo largo de los años. A mi hermano José Martin por mostrarme que incluso los hermanos menores tienen algo que enseñarte. A mis abuelos Juanita y Martin por sus grandes enseñanzas y ayuda cuando más se lo necesita. A mi demás familia que también me ha apoyado y ha compartido sus experiencias muy enriquecedoras.

Agradezco a mi tutor Pablo Padilla por brindarme la oportunidad de trabajar con él, por sus grandes enseñanzas tanto en el ámbito académico como en el ámbito personal. Por darme nuevas perspectivas en mi carrera y darme la oportunidad de incursionar en la ciencia. Admiro su gran conocimiento y capacidad para trabajar arduamente, así como su incalculable amistad y solidaridad con sus estudiantes y con las personas con quien labora. Y sobre todo la enorme paciencia que tiene para lograr transmitir sus conocimientos ya sea en esas clases del taller o en una tarde tomando un buen café.

También quiero agradecer a mi amiga Xhidenhi por brindarme esa infinita amistad que difícilmente se encuentra en estos tiempos modernos. A mis otros amigos Ema, Fausto, Dwyna, Hector, Daniela, Lalo, Isareli y Samantha que aunque haya pasado mucho tiempo, siempre están ahí para contarme alguna historia sobre su vida. A toda la gente que organiza las Escuelas de otoño de biología matemática que son una gran experiencia y una gran oportunidad para entablar amistad con personas inmersas en esta área no tan común de la ciencia en el país.

También quiero agradecer a personas que aunque nunca tuve contacto directo con ellas si tuvieron un impacto importante en mi vida, en mi pensamiento y en la forma en que veo el mundo: Akira Toriyama (mangaka y creador de Dragon Ball), Stephen Hawking (físico y divulgador de la ciencia), Paul Zaloom (actor del programa *El mundo de Beakman*) y Kurt Cobain (vocalista de la banda Nirvana).

Por ultimo agradezco a Carolina Barriga, Agustín Mercado, Miguel Lara y Eniak Hernández por el tiempo que me brindaron para revisar mi trabajo y los comentarios valiosos que hicieron, los cuales mejoraron enormemente este trabajo y sin estos no habría sido lo que es.

Contenido

Parte I

Introducción.....	8
Capítulo 1	
Aspectos biológicos de la sincronización	8
1.1 Ritmos biológicos	8
1.2 Osciladores biológicos	11
1.3 Sincronización por percepción de quórum	14
Capítulo 2	
Herramientas matemáticas para la modelación	18
2.1 Sistemas de ecuaciones diferenciales	18
2.1 Autómatas celulares.....	21
2.2 Teoría de juegos	24
Capítulo 3	
Modelación basada en agentes	28
3.1 Introducción a la modelación basada en agentes	28
3.2 Netlogo	32

Parte II

Metodología	36
4. Hipótesis.....	36
5. Objetivos Generales	36
5.1 Objetivos particulares:	37
6. Material y método	37
6.1 Análisis del modelo matemático y la matriz de pagos	38
6.2 Aprendizaje del uso del software	40
6.3 Protocolo ODD.....	40
6.3.1 Visión general y planificación	41
6.3.2 Diseño de conceptos	43
6.3.3 Detalles	45
6.4 Llevar a cabo las simulaciones.....	48

Parte III

Resultados.....	49
7. Resultados.....	49
7.1 Modelos en Netlogo	51

7.1.1 Modelo de sincronización en tres líneas celulares.....	51
7.1.2 Modelo de sincronización poblacional.....	56

Parte IV

Discusión y

Conclusiones	62
8. Discusión	62
9. Conclusiones	73
Referencias	75
Apéndice A: Código en Netlogo	78
Código del modelo poblacional en Netlogo	78
Apéndice B: Código en Polymath	
Interfaz de Polymath	88

Resumen

La sincronización siempre ha sido un tema que ha fascinado a las personas y en especial a los científicos por su continua presencia en muchos fenómenos que ocurren a nuestro alrededor. Desde los primeros astrónomos que se aventuraron a observar las estrellas y planetas encontraron patrones de movimiento que fueron sincronizados y ajustados a nuestras actividades para crear los primeros calendarios. Los físicos también observaron la sincronización de relojes mecánicos y utilizaron estos nuevos mecanismos para mejorar sus rutas de navegación. Los primeros viajeros occidentales que llegaron a los bosques del sur de Asia quedaron impresionados por las grandes congregaciones de luciérnagas que brillaban al unísono (Strogatz, 2003). La gente siempre queda fascinada por las ingeniosas coreografías que se realizan en la danza y en el teatro.

En física se tienen muy bien estudiados los sistemas que tienen una dinámica periódica u oscilante y se han desarrollado muchos modelos matemáticos que intentan explicar estos fenómenos de sincronización. Sin embargo en biología, hay una infinidad de fenómenos oscilantes y muchos de ellos se encuentran sincronizados en los organismos pluricelulares, incluso en nosotros mismos los seres humanos. Nuestro corazón, cerebro, nuestro sistema hormonal, nuestro ciclo de vigia y sueño, todo se encuentra regulado por miles y miles de osciladores que ajustan sus ritmos debido a algún tipo de interacción entre ellos.

Las pequeñas poblaciones de bacterias y microorganismos también sincronizan su actividad por diferentes mecanismos. Uno de los mecanismos más estudiados es el sistema de percepción por quorum. En el 2000 Elowitz y colaboradores diseñaron una construcción genética en *Escherichia coli* llamado el represilador de Elowitz (Elowitz *et. al*, 200) el cual tiene un comportamiento interesante ya que genera oscilaciones del tipo sinusoidales del mRNA y de las proteínas en las células. En un trabajo posterior ellos proponen un modelo matemático con el cuál puedan acoplar estos relojes moleculares sintéticos en toda una población bacteriana, incorporando el sistema de percepción por quorum a su construcción genética. Sin embargo analizar este tipo de sistemas matemáticos resulta demasiado difícil y consume mucho tiempo realizar simulaciones computacionales de tantas variables. Recientemente la modelación basada en agentes es una herramienta computacional muy importante, la cual permite desarrollar, analizar y experimentar modelos compuestos por agentes, los cuales interaccionan entre ellos y con su ambiente siguiendo un conjunto de reglas dadas por el creador del modelo. Por lo anterior el propósito de este trabajo es desarrollar un modelo basado en agentes que permita explorar el fenómeno de sincronización de bacterias empleando el sistema de comunicación por percepción por quorum.

El primer capítulo de este trabajo introduce al concepto de ritmo y sincronización en biología. Se mencionan algunos ejemplos de ritmos y se explica el sistema de percepción por quorum como medio de comunicación entre células bacterianas para lograr la sincronización de su oscilador interno. En el capítulo 2 se mencionan las herramientas matemáticas más importantes para la modelación de fenómenos biológicos y en el desarrollo de modelos computacionales. En el capítulo 3 se habla en detalle sobre las características de los modelos basados en agentes sobre todo del programa Netlogo que se empleó para el desarrollo de este modelo.

En La parte 2 se describe la metodología empleada para desarrollar un modelo computacional basado en agentes siguiendo el protocolo ODD propuesto por Railsback y colaboradores en el 2009. A su vez se describen las ecuaciones utilizadas para determinar la actividad del sistema de percepción por quorum aunque estas ecuaciones no se analizaron formalmente en el contexto matemático. En la parte 3 se describen y discuten los resultados obtenidos siendo 2 modelos los que se desarrollaron: un modelo que sigue la actividad de 3 líneas celulares y otro modelo donde se incorpora la densidad poblacional. Ambos modelos mostraron que la actividad del represilador no se altera a pesar del efecto de acoplamiento por percepción por quorum, sin embargo valores extremos en los parámetros utilizados si afectan enormemente el comportamiento del represilador así como de la fluorescencia. Se observó que el aumento población si incrementa los valores de inducción y por lo tanto mantiene a la población en un estado de fluorescencia más estable, pero no se encontró un patrón periódico entre los dos estados que hablara de una sincronización de las células. Sin embargo al comparar las gráficas simulaciones del sistema de ecuaciones de Elowitz y las gráficas obtenidas en el modelo de Netlogo, el comportamiento tanto de la actividad genética como de la fluorescencia y del autoinductor externo es muy parecido en ambos modelos. Esto sugiere que puede existir cierta periodicidad a nivel de la actividad genética de las estrategias génicas lo cual sería interesante analizar con otras herramientas matemáticas.

Parte I

Introducción

Capítulo 1

Aspectos biológicos de la sincronización

En este capítulo se describen algunas de las propiedades y características de la ritmicidad y la sincronización en el contexto biológico. Se mencionan ejemplos sobresalientes y trabajos previos que se han hecho en el campo de la cronobiología y la biología molecular. También se revisan los trabajos más recientes en el campo de la biología de sistemas que han permitido diseñar construcciones sintéticas de osciladores y además acoplarlos generando relojes moleculares sintéticos (Dunlap *et al.*, 2004; Koukkari y Sothorn, 2006; Goldbeter, 1996).

1.1 Ritmos biológicos

Los ritmos biológicos son fenómenos inherentes a la propia vida y estos se presentan en todos los niveles de organización de los seres vivos, desde los organismos unicelulares hasta los grandes ecosistemas, con periodos que van desde fracciones de segundos hasta años. Estos ritmos son dependientes de un reloj biológico endógeno y pueden ser perturbados por efectos de su ambiente ya sea que se trate de una sola célula o de un individuo como los seres humanos. El ritmo se define como un cambio que se repite periódicamente. En los fenómenos biológicos muchos de estos ritmos forman ciclos cuyos periodos pueden llegar a durar casi 24 horas, por lo

que se denominan ritmos circadianos. Otros periodos pueden durar menos de 20 horas por lo que se nombran como ritmos ultradianos (Tabla 1) y aquéllos que llegan a durar más de 28 horas, los llamados ritmos infradianos, pueden ser medidos en semanas, meses (circalunar), años (circanual) o en tiempos aún mayores (Koukkari & Sothorn, 2006).

Ritmo Biológico	Periodo
Ritmos neurales	0.001 a 10 segundos
Ritmos Cardiacos	1 segundo
Oscilaciones por calcio	Segundos a minutos
Oscilaciones bioquímicas	30 segundos a 20 minutos
Oscilador mitótico	10 minutos a 30 horas
Reloj de segmentación	30 minutos a 2 horas
Ritmos hormonales	10 minutos a 3-5 horas (incluso 24 horas)
Ciclo del sueño-vigilia	Cerca de 24 horas
Ciclo menstrual (circalunar)	28 días
Hibernación y estivación (circanual)	1 año
Ritmos en ecología y epidemiología	Varios años

Tabla 1. Lista de los principales ritmos biológicos clasificados de acuerdo a su periodo. Tomada de Goldbeter, 1996.

La disciplina que se encarga de estudiar estos ritmos biológicos es la cronobiología. Uno de los primeros en realizar pruebas experimentales sobre los ritmos biológicos fue el astrónomo francés Jean-Jacques Dortous de Mairan. Él al observar que el movimiento de apertura y cerrado de las hojas de la planta *Mimosa púdica* (Figura 1) se repetía día con día, se preguntó si esto era debido a la alternancia de luz y oscuridad en el lapso de 24 horas (Albrecht, 2010). De Mairan concluyo que “la planta al ser sensitiva debía sentir la presencia del Sol sin tener que verlo de alguna manera”, posteriormente concluyo que la planta debía de contener algún tipo de reloj endógeno que controlara dicho ritmo de apertura y cerrado de las hojas. Sin embargo también consideró que otros factores externos afectaban las variaciones en este movimiento de la planta, tales como los campos magnéticos de la tierra y la temperatura. Así muchos de los estudios sobre ritmos biológicos a principios del siglo XX siguieron esta corriente y se centraron en dilucidar la relación de la rotación axial del planeta que conducía a ciclos biológicos circadianos (Albrecht, 2010). Sin embargo no fue hasta mediados del siglo XX, gracias a los trabajos de Pittendrigh y Aschoff sobre la eclosión de las larvas de *Drosophila pseudoobscura*, que se establecieron las principales características de un reloj biológico: un periodo, con una duración sostenida y continua bajo condiciones constantes pero influenciada por señales ambientales tales como la luz y la temperatura (compensación o aclimatación) (Bass, 2012).



Figura 1. Planta de *Mimosa pudica* que presenta una respuesta al tacto contrayendo sus hojas como si se cerraran. Es un ejemplo bien estudiado de un ritmo circadiano vegetal regulado por la luz.

Actualmente se sabe que muchos de los mecanismos que producen algún tipo de ritmicidad en diferentes organismos están regulados principalmente por la síntesis de proteínas y mRNA (proteínas y genes reloj) en las células. Por ejemplo, en el caso de la mosca de la fruta (*Drosophila*) se han estudiado ampliamente los genes responsables de regular su ritmo circadiano, lo que la convierte en un modelo idóneo en genética para la investigación molecular de la regulación circadiana (Allada *et al.*, 2001). En general la base molecular de este modelo consiste en la expresión autosostenida, con un ritmo de 24 horas de los genes canónicos *tim* y *per* (Sehgal *et al.*) Sus productos proteicos actúan para reprimir la transcripción de sus propios genes en un asa de retroalimentación negativa. Dentro de esta asa se acoplan y desacoplan a activadores transcripcionales para formar un complejo dinámico multiprotéico o circadiasoma (Hardin *et al.*, 1990). Las características claves de la dinámica molecular es que hay un retraso entre la inducción de la transcripción de *per* y *tim* y por otro lado, la translocación hacia el núcleo de las proteínas represoras que estos genes codifican. Este retraso crea una separación temporal entre las fases de inducción y represión lo cual es necesario para generar las oscilaciones y de no existir este desfase de tiempos, los niveles transcripcionales llegarían a un equilibrio. Otra característica importante es que la vida media del mRNA de *per* y *tim* y sus productos proteicos es bastante corta, está bastante regulada y está muy bien adaptada para formar parte de este mecanismo de cronometraje (Wager-Smith y Kay, 2000).

La dinámica molecular del ritmo circadiano de *Drosophila* está regulado de la siguiente manera: durante el mediodía las proteínas CLK junto con CYC (de los genes *clock* y *cycle*) se unen a las secuencias E-box de los genes *per* y *tim* activando ligeramente su transcripción. De esta manera los niveles de mRNA *per* y *tim* comienzan a elevarse pero debido a la fosforilación por la proteína DBT (double-time) PER no se ve acumulada. Al caer la noche los niveles de TIM también

se han acumulado protegiendo a PER de la degradación, de tal forma que los heterodímeros de TIM-PER comienzan a formarse. Durante la noche estos heterodímeros son translocados hacia el núcleo, asociándose con CLK y CYC inhibiendo su habilidad para unirse con el DNA, así la producción del mRNA de *per* y *tim* disminuye. El amanecer estimula al fotoreceptor CRY para secuestrar las proteínas TIM y degradarlas por la vía de degradación ubiquitina-proteosoma. Y para el mediodía, estos eventos causan un cambio en el balance de un exceso de PER-TIM en el núcleo por un exceso, ahora, de CLK- CYC preparando a la mosca para un nuevo ciclo de estos componentes genéticos (Wager-Smith y Kay, 2000).

Estos y muchos otros estudios moleculares han permitido desarrollar modelos más realistas en términos de precisión bioquímica, con aplicaciones en diferentes campos de la medicina y de las ciencias biológicas (Goldbeter, 1996). Diferentes pruebas diagnósticas en el campo de la medicina se ven afectadas por los ritmos circadianos (Wever, 1979). Por ejemplo, las pruebas para diagnósticos de hipertensión pueden arrojar resultados con mayor variación si éstas se realizan en horarios nocturnos en vez de matutinos. Esto sucede también en las pruebas de tolerancia a la glucosa, las cuales, ayudan a obtener un diagnóstico más preciso para tratar los distintos tipos de diabetes. De tal forma que han surgido nuevas áreas de estudio como la cronofarmacología, la cronopatología y la cronotoxicología, que se encargan de estudiar la manera y el grado en que la cinética y la dinámica de los medicamentos son afectadas directamente por los ritmos biológicos endógenos y también, cómo el tiempo de dosificación afecta el cronometraje biológico y las características (periodo, amplitud y fase) de los ritmos biológicos (Smolensky & Peppas, 2007).

1.2 Osciladores biológicos

Muchos sistemas bioquímicos y celulares presentan comportamientos oscilatorios complejos, para lo cual los modelos matemáticos arrojan luz sobre algunas de las condiciones que generan estos comportamientos caóticos. En la década de los 60's y 70's hubo grandes esfuerzos por resolver la cuestión de que las oscilaciones relacionadas con los ritmos dentro de las células, tuvieran su origen en la síntesis de enzimas cómo resultado de la regulación genética. Lo anterior dio la oportunidad para abordar de forma experimental y teórica la cuestión de las bases moleculares de un ritmo biológico, cuyo origen difiera del comportamiento de las ya conocidas en ese entonces, oscilaciones eléctricas presentes en las neuronas (Goldbeter, 2002).

A nivel genético varios ensayos experimentales han mostrado que las interacciones regulatorias entre genes resultan en estados biestables u oscilaciones. Las oscilaciones se definen como la continua transición entre dos estados y esto se puede dar cuando el equilibrio dinámico de un sistema es perturbado y distintas fuerzas de restauración tratan de regresar al sistema a su estado de equilibrio (Aubel & Fussenegger, 2010). Las oscilaciones están presentes no sólo a nivel celular, en nuestros órganos y en los diferentes procesos fisiológicos de nuestro cuerpo también existen fenómenos oscilatorios. A nivel poblacional como en las interacciones depredador-presa de diferentes especies, en la actividad migratoria de aves y peces e incluso, en la economía del mercado y las finanzas también hay procesos oscilatorios inmersos (Mishchenko, 2014). Además han sido utilizadas y explotadas en diversas áreas de la ingeniería como la mecánica, la electrónica y la óptica para generar distintos productos que empleamos en la vida cotidiana (relojes de péndulo, chips computacionales, lasers, etc.)

Las oscilaciones observadas *in vitro* en el sistema glicolítico del músculo y en las células de levaduras fueron y aún son el prototipo para estudiar las oscilaciones bioquímicas como resultado de la regulación de la actividad enzimática (Mads, *et al.*, 2005). Estas oscilaciones se producen debido a una propiedad bastante curiosa de la enzima fosfofructocinasa (PFK) la cual cataliza la fosforilación de la de la fructosa-6-fosfato a fructosa-1,6-bifosfato utilizando ATP como el donador de fosfato. A pesar de que la PFK consume ATP, esta ruta glicolítica produce más ATP del que consume. Así para regular apropiadamente la producción de ATP, el ATP inhibe a PFK mientras que el ADP la activa. Si hay un enriquecimiento energético celular (ATP alto y ADP bajo) PFK se inhibe y el flujo de los azúcares en la ruta glicolítica se ve detenido. Mientras que si los niveles de ATP caen y los niveles de ADP suben, la PFK se activa y la glicolisis comienza de nuevo. En principio este sistema de retroalimentación negativo estabiliza el suministro energético de la célula al generar estas oscilaciones en los niveles de ATP (Goldbeter, 1996).

Siguiendo los estudios clásicos de Jacob y Monod, Goodwin fue uno de los primeros en proponerse investigar, teóricamente, las condiciones con las cuales la retroalimentación negativa, en forma de represión transcripcional, pueden producir oscilaciones en la síntesis de proteínas (Goldbeter, 2002). En 1965 Goodwin propuso un modelo matemático de un oscilador genético que consta de un solo gen, por lo que fue una base importante para el estudio teórico de la generación de oscilaciones en las células. En este modelo, la síntesis periódica de un enzima es regulada por un asa de retroalimentación negativa vía represión transcripcional del gen de la enzima por un inhibidor (Ingalls, 2013). El mRNA, denotado como X, se traduce a una enzima Y, que produce un metabolito Z (factor de transcripción) que inhibe la expresión del gen X (Figura 2). De tal manera que este sistema hace uso de la retroalimentación negativa para generar oscilaciones de las concentraciones del mRNA y de la proteína.

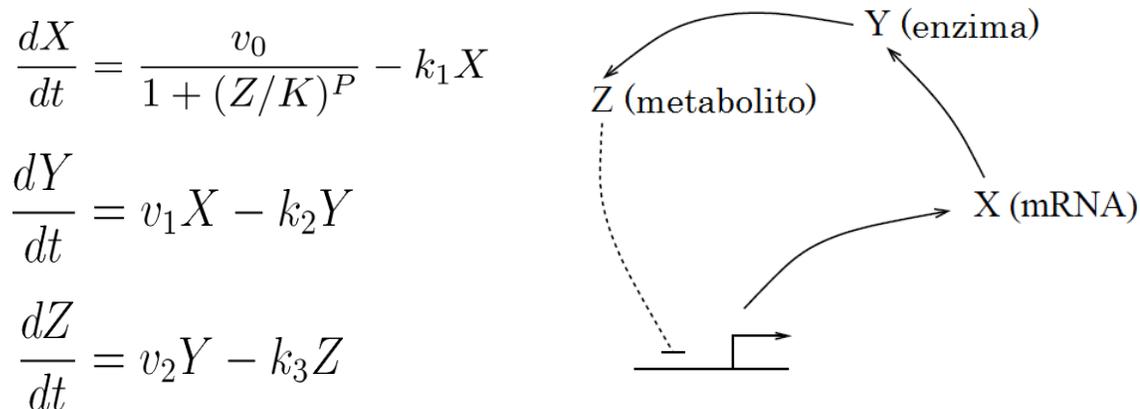


Figura 2. Ecuaciones y esquema de la dinámica del Oscilador de Goodwin.

En las ecuaciones de la figura 2, las variables X , Y y Z corresponden a las concentraciones de mRNA, enzima, y producto proteico respectivamente; v_0 , v_1 y v_2 determinan las tasas de catálisis, transcripción y traducción; k_1 , k_2 y k_3 son constantes de velocidad para la degradación de cada componente; $1/K$ es la constante de unión del producto proteico al factor de transcripción; y p es una medida de la cooperatividad del represor. Analizando este modelo se observa que para obtener las oscilaciones bioquímicas, la cooperatividad de la retroalimentación negativa debe ser bastante alta, con una $p > 8$ y la tasa de degradación (k_1 , k_2 y k_3) de los tres componentes deben ser casi iguales entre ellos (Dubitzky et al, 2013).

Más recientemente en el campo de la biología molecular y la biotecnología se han diseñado varios osciladores sintéticos los cuales han servido como herramientas clave para predecir y entender las propiedades reales de la dinámica celular de las redes genéticas oscilatorias (Singh, 2013). El primer oscilador molecular sintético fue diseñado por el grupo de Michael Elowitz en la universidad de California a principios del año 2000 y fue probado experimentalmente en la bacteria *Escherichia coli*. La construcción genética consta de tres módulos diseñados mediante ingeniería genética y organizados de forma cíclica generando un asa de retroalimentación negativa parecido al juego de piedra-papel y tijeras. Piedra inhibe a tijeras, pero tijeras inhibe a papel, así que si tijeras está inhibido papel se mantiene activado. Luego papel inhibe a piedra por lo que si piedra está inhibido tijeras se vuelve a activar. Esta construcción genética esta acoplada a otra construcción reportera que expresa una proteína verde fluorescente (GFP) y su expresión está controlada por la actividad de *tetR*. Esta configuración genera una expresión rítmica de esta proteína fluorescente en poblaciones en crecimiento de *E. coli*. El Represilador (Figura 3) es una red de tres genes: *lacI* (del operon del transporte de lactosa en *E. coli*), *tetR* (del transposón de resistencia a la tetraciclina Tn10) y *cl* (del vector de clonación del fago lambda), cuyos productos inhiben la transcripción de uno al otro de forma cíclica. El gen *lacI* expresa la proteína LacI, que inhibe la transcripción del gen *tetR*. Luego su producto, TetR, inhibe la transcripción del gen *cl* y el

producto de este último gen (CI), a su vez inhibe la expresión de *lacI*, completando el ciclo. Este sistema se detecta por medio de otra construcción genética (reportero) que expresa la proteína fluorescente GFP y que está regulado por el promotor de *tetR* (Elowitz & Leibler, 2000).

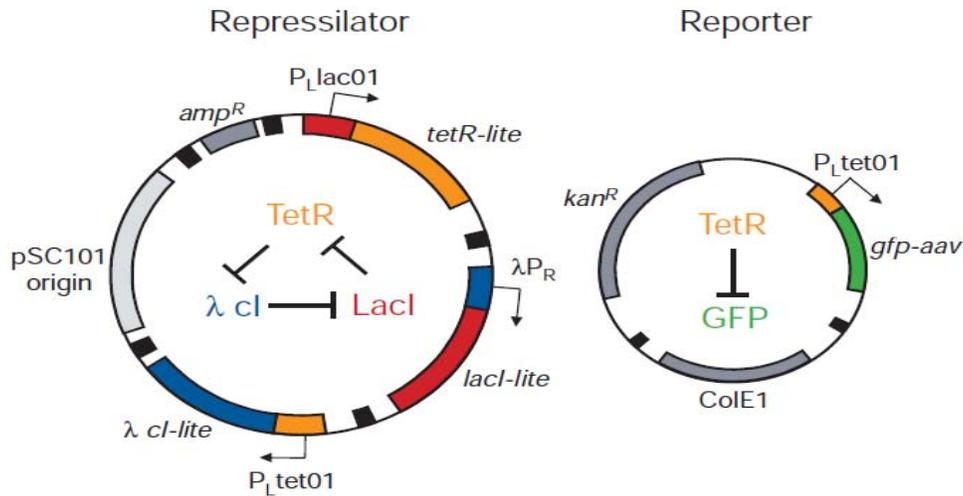


Figura 3. Construcción genética del represilador de Elowitz (lado izquierdo) como un asa de retroalimentación negativa cíclica compuesta por tres genes represores con sus correspondientes promotores (P_Llac001, λP_R y P_Ltet001). Otros componentes son: origen de replicación (pSC101) y gen de resistencia a la tetraciclina (amp^R). Plásmido reportero (lado derecho) que expresa la proteína GFP controlada por el promotor de *tetR*. Modificada de Elowitz & Leibler, 2000.

1.3 Sincronización por percepción de quórum

La sincronización es un proceso fundamental de los fenómenos no lineales. La esencia de la sincronización se puede explicar de forma simple: objetos oscilantes débilmente acoplados ajustan sus ritmos y comienzan a oscilar con frecuencias comunes (armónicas) y con fases estrechamente relacionadas. Esta descripción cualitativa se remonta a los experimentos de Christian Huygens a mediados del siglo XVII en Holanda considerándose el fenómeno no lineal más viejo descrito de manera científica, por lo que sigue atrayendo la atención de científicos de las ciencias naturales y la ingeniería. La sincronización es el caso más evidente de un efecto general llamado seguimiento de fase: muchos osciladores van siguiendo el mismo patrón y al mismo paso. Cuando dos osciladores idénticos están acoplados se presentan dos posibilidades: sincronía (una diferencia de fase de cero) y antisincronía (una diferencia de fase mayor a 0) (Strogatz & Stewart, 1993). Por ejemplo en la locomoción bípeda de un canguro sus patas posteriores oscilan periódicamente y ambas tocan el suelo en el mismo instante, en cambio en la locomoción de un humano sus piernas tocan el suelo alternativamente (Figura 4).

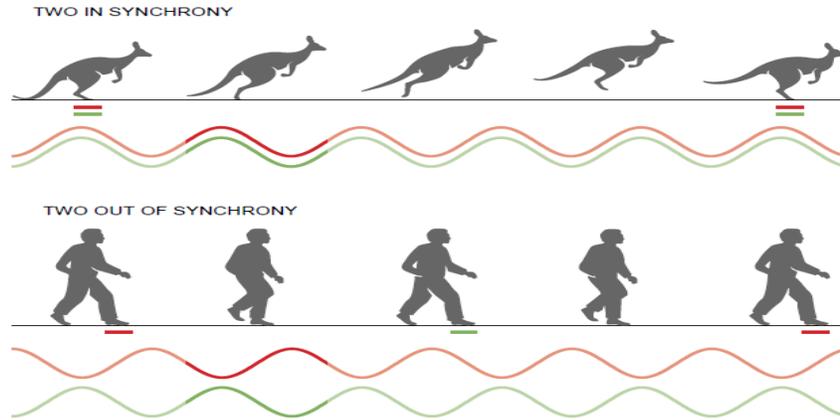


Figura 4. Formas en que se pueden acoplar dos osciladores: de forma sincrónica (arriba) en la locomoción del canguro o antisincrónica (abajo) en la locomoción humana. Modificada de Strogatz & Stewart, 1993.

Sin embargo la sincronización en el caso de poblaciones de organismos no es tan sencilla. El problema con las células es que, aunque las frecuencias y amplitudes de las oscilaciones en las células individuales pueden ser programadas genéticamente, cada célula sigue su propio tiempo interno. El caso de las poblaciones bacterianas tampoco es la excepción. Hasty y colaboradores tomaron el sistema de comunicación bacteriano conocido como percepción por quórum para lograr la sincronía deseada en una generación de osciladores moleculares (Danino *et al.*, 2010). El sistema de percepción por quórum bacteriano produce, libera y detecta pequeñas moléculas llamadas autoinductores, lo que genera una respuesta a esta señal bioquímica. A medida que una población de bacterias productoras de autoinductor crece, la concentración de autoinductor liberado también se incrementa. Cuando se alcanza un umbral crítico de concentración de la molécula señalizadora, la bacteria es capaz de actuar debido a su presencia e iniciar una cascada de señalización que resulta en cambios en la expresión de genes blanco. De tal forma que este sistema de percepción por quórum le permite a las bacterias coordinar la expresión génica de toda la comunidad, lo que es bastante útil e interesante ya que funcionan como un organismo multicelular (Rosenberg *et al.*, 2013).

Así esta forma intercelular de acoplamiento global les permitió a Hasty y colaboradores diseñar un oscilador sincronizado en base en los elementos de la maquinaria de percepción por quorum de *Vibrio fischeri* y *Bacillus thuringiensis*. En la naturaleza existen gran variedad de sistemas de percepción por quorum (Reading *et al.*, 2006) pero el equipo de Hasty empleó el sistema de los genes *luxI*, *aiiA* e *yemGFP* los cuales se construyeron bajo el control de tres copias idénticas del promotor de *luxI*. La enzima de *luxI* es una sintasa que produce una acil-

homoserinalactona (AHL), que es una molécula que puede difundir a través de la membrana celular y que se une intracelularmente a LuxR, que se expresa de forma constitutiva. El complejo formado por LuxR y AHL es un activador transcripcional para el promotor de *luxI*. *AiiA* es una proteasa que regula negativamente a este promotor al catalizar la degradación de AHL (Figura 5). De esta forma las oscilaciones ocurren ya que la pequeña molécula de AHL tiene un papel dual: permitir la activación de los genes necesarios para generar las oscilaciones intracelulares y mediar el acoplamiento entre células. Las oscilaciones se pueden observar por la fluorescencia de la proteína verde fluorescente (*yemGFP*) (Danino *et al.*, 2010).

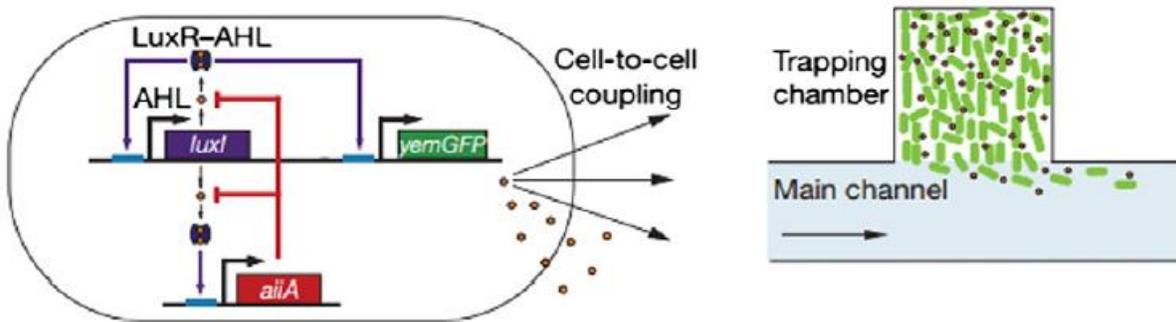


Figura 5. Diagrama de la red del reloj genético sincronizado. Del lado izquierdo están los tres genes en tres módulos transcripcionales. Del lado derecho el dispositivo de microfluidos utilizado para mantener una densidad constante de bacterias de *E. coli*. (Modificada de Danino *et al.*, 2010).

En el 2004 Elowitz y colaboradores también propusieron un modelo matemático para acoplar una población de bacterias por medio del sistema de percepción por quórum. En su modelo ellos proponen incorporar este aparato de comunicación intercelular al represilador, poniendo al gen que codifica a *luxI* bajo el control de la proteína del represilador, *Lacl*. Adicionalmente, una segunda copia del gen del represilador (*lacl*) se insertó dentro de la maquinaria genética de *E. coli* de tal manera que su expresión es inducida por el complejo LuxR-AI (Figura 6). El resultado es la aparición de un asa de retroalimentación en el represilador, la cual se refuerza si los niveles de *Lacl* son más similares entre las células vecinas. Así conforme la densidad celular se incrementa, la difusión de las moléculas extracelulares (AI) hacia las células, provee un mecanismo de acoplamiento intercelular, lo cual lleva a un seguimiento parcial de la frecuencia de las células. Los resultados de sus simulaciones mostraron que la transición de un estado no sincronizado a un estado sincronizado, existe conforme la fuerza de acoplamiento se incrementa (lo cual es causado por un incremento en la densidad celular (Garcia-Ojalvo *et al.*, 2004). De tal forma que ellos concluyen que este diseño puede ser incorporado a las células que contengan su

sistema de represilador permitiendo la sincronización de un ensamble de represiladores sintéticos. Sin embargo no se ha podido llevar a cabo la comprobación experimental de este modelo que permita el acoplamiento de las colonias bacterianas con dicha construcción genética.

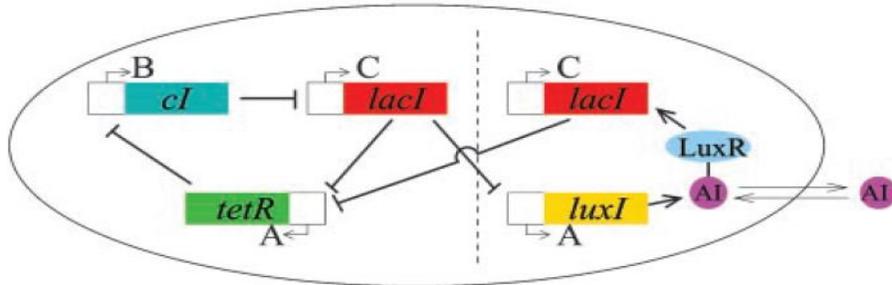


Figura 6. Esquema de la red del represilador acoplado por medio del sistema de percepción por quórum. El módulo del represilador original está a la izquierda de la línea punteada y el nuevo módulo para el acoplamiento está del lado derecho. Modificado de Garcia-Ojalvo, *et al.* 2004.

Una forma de estudiar los ritmos biológicos es por medio de los modelos matemáticos y la biología computacional (Lema *et al.*, 2001). La modelación permite abordar nuevas formas de investigación cuando la intuición alcanza sus límites. Esto sucede comúnmente cuando se estudian procesos celulares que involucran un gran número de variables que se acoplan por varias interacciones regulatorias. Así los modelos matemáticos nos dan rangos bastante precisos de los parámetros que dan lugar a oscilaciones sostenidas y las simulaciones computacionales permiten determinar de forma rápida los efectos cualitativos y cuantitativos de cada parámetro. Lo anterior ayuda a identificar parámetros clave que tienen los efectos más profundos en la dinámica de nuestro sistema (Goldbeter, 2002).

Capítulo 2

Herramientas matemáticas para la modelación

En este capítulo se dan los conceptos matemáticos necesarios así como una breve reseña histórica para plantear un modelo que nos ayude a entender el fenómeno de acoplamiento de relojes moleculares sintéticos. Se explicarán los fundamentos de los sistemas de ecuaciones diferenciales, los autómatas celulares y la teoría de juegos como los métodos matemáticos necesarios para desarrollar un modelo que genere un comportamiento de sincronización del represilador diseñado por Elowitz (Deutsch & Dormann, 2005; Wolfram, 1984,2006; Ermentrout & Edelstein-Keshet, 1993; Adamatzky, 2010).

2.1 Sistemas de ecuaciones diferenciales

Los fenómenos biológicos son en gran medida sistemas dinámicos. Siempre hay fuerzas naturales que actúan para cambiar el comportamiento de una célula, un organismo, una población o incluso un ecosistema. Dentro de los sistemas dinámicos se distinguen dos clases de modelos: los modelos determinísticos y los modelos estocásticos. Determinístico se refiere a que el futuro está enteramente predicho (determinado) por el mismo modelo. Estocástico se refiere a que eventos aleatorios (estocásticos) afectan al sistema biológico, por lo cual, el modelo solamente puede predecir la probabilidad de que uno o varios hechos ocurran en el futuro (Otto & Day, 2007).

La construcción de los modelos matemáticos comienza con una pregunta que se hace sobre un fenómeno estudiado, ya sea biológico o de cualquier otra índole y se intenta plasmar esa pregunta empleando el lenguaje matemático. Una vez que se ha razonado el problema, el primer paso es describir las variables o las entidades que cambian en el tiempo en nuestro modelo. Éste primer paso es sumamente importante porque sintetiza lo que es verdaderamente relevante del fenómeno biológico y descarta muchos otros procesos que no afectan el comportamiento biológico y que vuelven más complejo nuestro modelo. El número de variables dependerá, por lo tanto, del problema en cuestión. Por ejemplo, si lo que queremos saber es cómo cambia la población de organismos en una clásica interacción depredador-presa lo que nos va interesar es: el número de depredadores que hay, el número de presas, cuantas presas son devoradas en cierto intervalo de tiempo y como contribuye eso al desarrollo de la población del depredador (Smith & Smith, 2007.). En este ejemplo no nos interesa saber cómo es la variabilidad genética de ambas poblaciones, cómo es su ciclo de vida, ni siquiera nos interesa saber la edad o el sexo de los individuos ya que la tasa de depredación es la misma independientemente de estos factores. Al

descartar todas estas variables mejoramos el modelo al plasmar lo que realmente afecta a la depredación y al comparar el modelo con los datos experimentales, vemos que efectivamente el modelo predice y describe el comportamiento de ambas poblaciones. Por lo tanto, los modelos deben ser lo más simples posibles pero no más simples parafraseando a Einstein. Como consejo, sólo se deben añadir más variables cuando el modelo no consigue acercarnos al problema en cuestión (Otto & Day, 2007).

Las variables por lo general se representan por letras, por ejemplo x o y para dos poblaciones diferentes. Sin embargo es buena idea escoger las letras de tal forma que ayude a recordad las variables que representan. En el ejemplo de la interacción depredador-presa la población de presas puede representarse con la letra p y la de depredadores con la letra d . Si nuestras variables cambian en el tiempo es útil emplear la notación $p(t)$ lo que nos dice que nuestra variable está en función de algo más (tiempo en este caso) y se lee como “ p en el tiempo t ”. Una vez que se tiene una lista preliminar de las variables el siguiente paso es determinar la dinámica del modelo, para describir los cambios en estas variables. Como ya se mencionó los modelos discretos asumen que las variables cambian de una unidad de tiempo (por ejemplo días, años o generaciones) a otra, mientras que los modelos continuos asumen que las variables cambian en cualquier punto en el tiempo con incrementos o decrecimientos ocurriendo dentro de intervalos muy pequeños. Dependiendo del proceso biológico que estemos estudiando nuestro modelo será discreto (por ejemplo el desarrollo de nuevas ramas en un árbol) o continuo (crecimiento bacteriano en un biorreactor) siempre tomando en cuenta el contexto biológico en el que lo estemos analizando. En los modelos discretos, por lo general, se hace uso de las ecuaciones de recurrencia las cuales describen el valor de la variable (por ejemplo p) en la siguiente unidad de tiempo como una función de la variable en la unidad de tiempo actual:

$$p(t + 1) = \text{”alguna función de } p(t)\text{”}.$$

Tales ecuaciones se llaman recursivas porque uno puede aplicarlas a sí mismas recursivamente para encontrar como la variable cambia a través de un sinnúmero de unidades de tiempo (de t a $t+1$, luego de $t+1$ a $t+2$, etc.). Una forma equivalente de seguir a la variable es empleando ecuaciones en diferencias. Estas ecuaciones especifican qué tanto cambia una variable de una unidad de tiempo a la siguiente, y es simplemente la diferencia entre el valor obtenido en la ecuación de recursión para $p(t+1)$ y el valor actual de la variable $p(t)$:

$$\Delta p = p(t + 1) - p(t) = \text{”alguna función de } p(t)\text{”}.$$

En los modelos continuos, las ecuaciones que especifican la tasa de cambio de las variables a través del tiempo se llaman ecuaciones diferenciales. Este tipo de ecuaciones surgen

naturalmente en los modelos biológicos continuos ya que por lo general esperamos que la tasa de cambio de una variable, esté en función de su valor actual (Otto & Day, 2007):

$$\frac{d(p(t))}{dt} = \text{”alguna función de } p(t)\text{”}.$$

Posteriormente es necesario describir los parámetros del modelo. Estas son cantidades que influyen en la dinámica del modelo, pero permanecen fijas a través del tiempo mientras cambia la variable. Pueden ser valores positivos o negativos dependiendo de la realidad biológica, si contribuyen o no al incremento de la variable. Una vez que se tienen tanto las variables como los parámetros y se ha decidido qué tipo de modelo (discreto o continuo) se empleará, es hora de desarrollar las ecuaciones incorporando la dinámica de cambio de las variables. Conceptualmente hablando, la dinámica de las ecuaciones sigue todos los factores que causan que una de las variables se incremente o disminuya a través del tiempo y por lo general tiene la forma de:

$$p(t + 1) = n(t) + \text{incremento} - \text{decremento} \quad (\text{ecuación de recursión})$$

$$\Delta p = \text{incremento} - \text{decremento} \quad (\text{ecuación en diferencia})$$

$$\frac{d(p(t))}{dt} = \text{tasa de incremento} - \text{tasa de decremento} \quad (\text{ecuación diferencial})$$

Considere el siguiente ejemplo de la propagación de la gripe. Tenemos dos poblaciones o variables: aquellas personas que están contagiadas $n(t)$ y las personas que son susceptibles de contagiarse $s(t)$. El modelo asume que individuos con la gripe interactúan con los individuos susceptibles mediante una tasa de probabilidad de c al día, dando como lugar al número total de contactos por día por individuo como $c s(t)$. Dentro de los individuos infectados $n(t)$ de la población, se espera que ocurran un total de $c n(t) s(t)$ contactos por día a través de toda la población. La probabilidad de que uno de los contactos resulte en la transmisión del virus de la influenza es a , y por lo tanto se espera un total de $a c n(t) s(t)$ nuevos casos de influenza al día. De tal forma que las ecuaciones de recursión que muestren los cambios en la población de personas susceptibles y enfermas en el siguiente tiempo son las siguientes:

$$n(t + 1) = n(t) + a c n(t) s(t),$$

$$s(t + 1) = s(t) - a c n(t) s(t).$$

Nótese que en la ecuación de la población de personas susceptibles el término a se vuelve negativo ya que este se refiere al número de casos nuevos de influenza, por lo tanto contribuye negativamente al aumento de la población de personas susceptibles. Ahora para determinar cómo

cambian estas variables a través del tiempo es necesario desarrollar las ecuaciones diferenciales desarrollando los elementos que contribuyen al incremento o decrecimiento de las 2 poblaciones. Como se vio en las ecuaciones de recursión la población de personas infectadas está determinada por el número de contactos que tenga ésta con las personas susceptibles y la probabilidad de contagiarlas, por ello la ecuación queda como:

$$\frac{d(n(t))}{dt} = a c n(t) s(t)$$

Mientras que para la población de personas susceptibles el éxito de que se dé un contagio afecta negativamente a la población por lo tanto la ecuación queda como:

$$\frac{d(s(t))}{dt} = -a c n(t) s(t)$$

Una vez ya obtenido el sistema de ecuaciones que expliquen la dinámica de nuestro sistema es necesario analizar nuestro modelo. Hay diferentes maneras de analizar las ecuaciones, por ejemplo; análisis gráfico, realizar simulaciones, análisis de la estabilidad y el equilibrio, derivar soluciones generales, analizar el modelo por comportamiento periódico, etc. Una vez analizado y observado los resultados, si estos no tienen sentido o si no resultaron como uno esperaba, es necesario revisar de nuevo las ecuaciones y ver que puede ser lo que esté fallando. Rara vez la primeras propuestas de ecuaciones resultan en el modelo final de nuestro fenómeno biológico. Incluso cuando estamos seguros que no hay ningún error y que tanto los parámetros como nuestras variables son fieles a los aspectos biológicos reales (es decir que las unidades de las variables y parámetros correspondan a lo esperado biológicamente, si hablamos de decrecimiento que sean parámetros negativos, etc.) es posible que en realidad nuestra intuición sea lo que esté fallando. Puede que nuestro modelo sea muy simple o muy complejo pero esa es la parte importante en el proceso de modelación, que nos permite corregir y refinar nuestro entendimiento del sistema biológico (Otto & Day, 2007).

2.2 Autómatas celulares

La naturaleza ha creado varios sistemas biológicos los cuales se vuelven sistemas bastante complejos cuando queremos estudiarlos y analizarlos. Estos sistemas no pueden ser modelados cuantitativamente de forma precisa, por lo que los modelos matemáticos cuantitativos que se intentan emplear, se vuelven problemas numéricos muy complejos. Así que una forma de simplificar estos sistemas numéricos comúnmente intratables, es imitando las leyes físicas que las

rigen por una serie de reglas simples que son fáciles de computar de forma rápida y práctica. Los autómatas celulares (AC) son ejemplos de esta forma de simplificar los sistemas complejos que existen en la naturaleza para poder analizarlos cuantitativa y cualitativamente.

La noción de AC se originó con los trabajos de John von Neumann y Stanislaw Ulam a mediados de los años de 1940 donde trataban de responder el siguiente problema: “¿cuál será el tipo de organización lógica necesaria para que sea posible que un autómata (visto como un dispositivo artificial) se reproduzca a sí mismo? Uno de los primeros intentos de construir un autómata biológico fue el pato mecánico de Jacques de Vaucanson en 1735, el cual no solamente podía mover su cabeza, alas y cola sino que también tenía la capacidad de alimentarse (Figura 7), aunque no podía reproducirse (Deutsch y Dormann, 2005).

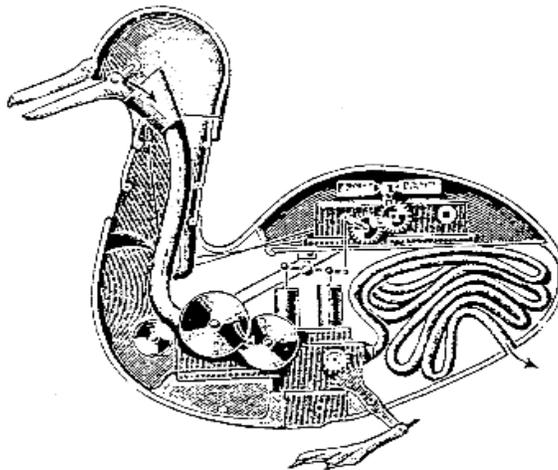


Figura 7. Esquema del diseño del pato mecánico de Vaucanson el cual podía mover la cabeza, las alas y la cola e incluso podía ser alimentado. Modificada de Deutsch y Dormann, 2005.

El primer modelo de auto-reproducción fue propuesto por von Neumann observando la naturaleza en especial la información genética contenida en el DNA generando una interpretación sintáctica y semántica de este fenómeno biológico. El autómata actualmente consiste de 2 partes: una construcción que es flexible y una unidad instruccional semejando a la dualidad entre computadora y programa, o alternativamente, a la célula y su genoma. De esta forma von Neumann se anticipó al desarrollo del código genético seguido del descubrimiento de la estructura de doble hélice del DNA por Watson y Crick, lo cual también correspondería a los procesos moleculares de traducción y transcripción en la célula. Sin embargo fue Ulam Stanislaw quien sugirió una “perspectiva celular” y contribuyó con la idea de restringir los componentes a celdas espaciales discretas distribuidas en una laticidad regular (Deutsch & Dormann, 2005).

En años recientes las ideas propuestas por John von Neumann y Stanislaw Ulam se han extendido a diferentes aplicaciones, pero en general los autómatas celulares (AC) se especifican por las siguientes características: una red discreta regular (L) de celdas (nodos, puntos) con sus condiciones de frontera; un conjunto de estados, generalmente pequeño y finito (E), que caracterizan a las células; un conjunto finito de celdas que define la interacción del vecindario (NI) de cada celda; una regla de transición (R), que determina la dinámica de los estados de las celdas.

A su vez los autómatas celulares pueden ser diseñados en una, dos, tres o más dimensiones dependiendo de las interacciones que tenga cada celda con su vecindario. El vecindario de dos dimensiones más popular es el vecindario de von Neumann. Este vecindario tiene, para una sola celda, 4 interacciones con 4 celdas adyacentes (Figura 8a). A cada celda r se le puede asignar un valor de estado $s(r)$ el cual es escogido generalmente de un conjunto finito de estados elementales ε . Los elementos de ε pueden ser números, símbolos u otros objetos, por ejemplo, células biológicas (Figura 8b). La dinámica del automaton (plural de autómatas celulares) está determinada por una regla local de transición R la cual va a especificar el nuevo estado de una célula como función de la interacción que tiene en el vecindario (Figura 8c).

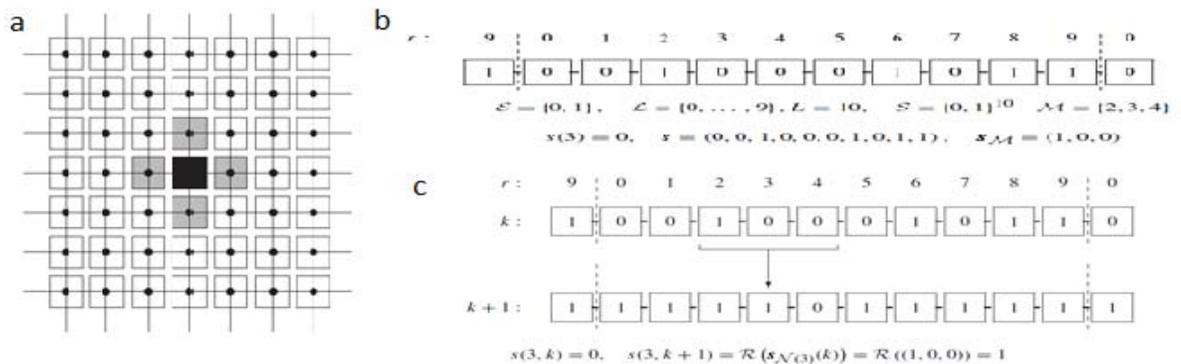


Figura 8. Representación gráfica del vecindario de von Neumann (a). Un vecindario unidimensional de un tamaño $L=10$, donde se tienen dos valores de estado $\varepsilon = \{0,1\}$, y el conjunto global de estados se denota como $s = \{0,0,1,0,0,0,1,0,1,1\}$ (b). Regla de transición se puede definir como: una celda se mantiene vacante si todos los vecinos se encuentran vacantes, mientras que si una celda vacante está rodeada por al menos 1 celda ocupada esta celda también se verá ocupada (c). Modificada de Deutsch y Dormann, 2005.

Los autómatas celulares se han empleado para crear modelos tanto en el campo de la física como en el de la biología y en este último ha tenido su aplicación principalmente en la ecología, la epidemiología, la biología evolutiva, y la bioquímica (Ermentrout y Edelstein-Keshet, 1993). Muchos de estos modelos representan las abstracciones de las ecuaciones diferenciales parciales que pretenden modelar los patrones espaciales y temporales de los fenómenos biológicos. Por lo que estos tipos de modelos de los autómatas celulares caen dentro de la clasificación de determinísticos, es decir, que una vez que se conocen las condiciones iniciales del sistema todos los estados subsecuentes pueden determinarse llevando a cabo las iteraciones de los estados anteriores (Deutsch y Dormann, 2005).

2.3 Teoría de juegos

Von Neumann utilizaba el póker y otros juegos colectivos para crear estructuras matemáticas maravillosas para el análisis matemático en diversos tipos de interacciones sociales, en las cuestiones militares y en la economía de las sociedades. Sin embargo no fue hasta el trabajo de Martin Novak que las ideas de la teoría de juegos se juntaron con las ideas de los autómatas celulares, al analizar una población de individuos hipotética en un mundo artificial parecido al juego del ajedrez. Aquí se encontró que la estrategia de los individuos está predeterminada, ya sea que siempre cooperen entre ellos o se nieguen a cooperar en lo absoluto. Así la población en este mundo artificial evolucionaba siguiendo una regla de transición inequívocamente determinada, lo cual permite observar la composición de la población indefinidamente siguiendo la misma dinámica de los autómatas celulares (Weibel, 2005).

Ha sido sólo hasta recientemente que ha surgido la idea de aplicar la teoría de juegos al campo de la biología, en particular a la teoría evolutiva de la Selección Natural, de ahí que surjan los primeros juegos evolutivos. Los modelos en los juegos evolutivos están aplicados a grandes poblaciones, donde los juegos son (usualmente) jugados entre los miembros de la población escogidos aleatoriamente, lo cual lleva a diferentes tipos de soluciones que pueden ser resueltas usando los juegos convencionales que ofrece la teoría de juegos. La primer aplicación explícita de la teoría de juegos a la biología evolutiva fue el trabajo de R.C Lewontin en “Evolution and The Theory of Games” en 1961 (Lewontin, 1961). Su trabajo involucró que las especies y no los individuos, fueran los que jugaran contra la naturaleza incluso si una estrategia particular es beneficiosa para las especies como un todo, si estas a su vez, pueden ser invadidas por otra estrategia alternativa eliminando la anterior estrategia, de tal modo solo una estrategia puede persistir en la población (Broom y Rychtar, 2013). Sin embargo John Maynard Smith y George Price

contribuyeron a establecer e independizar la teoría de juegos en la biología evolutiva así como introducir el que pensamiento poblacional gracias a su idea de las Estrategias Evolutivamente Estables (ESS) que va a la par con la teoría del equilibrio de Nash (Nowak, 2006).

Tradicionalmente la teoría de juegos analiza la interacción entre dos jugadores. La pregunta es ¿cómo se puede maximizar la rentabilidad en un juego, si un jugador no sabrá lo que va hacer el segundo? El concepto de racionalidad entra en juego. Un jugador puede asumir que el otro va actuar con el fin de maximizar su recompensa. Teniendo en cuenta esta premisa, entonces el otro jugador hará lo posible para maximizar también su recompensa. La teoría de juegos evolutiva no se basa en la racionalidad. En su lugar se considera una población de jugadores que interactúan en un juego donde los individuos tienen estrategias fijas y estos interactúan al azar con otros individuos. Estas interacciones tienen un costo denominado aptitud biológica, y el éxito en el juego se traduce como éxito reproductivo. Así que las estrategias que resultan mejores hacen que la reproducción sea más rápida, mientras que las peores estrategias hacen que las especies no puedan competir con las otras (Nowak, 2006). La teoría de juegos es idealmente adecuada para tratar los problemas de la biología evolutiva. Por ejemplo, las acciones de un individuo de una población están en función de lo que hagan los otros individuos de dicha población, así, esto se hace aplicable a uno de los mayores problemas encontrados en la genética de poblaciones clásica: la selección dependiente de la frecuencia (Rice, 2004).

La idea clave de la teoría de juegos evolutiva es considerar una población de jugadores A y B y tratar de igualar el costo de las estrategias empleadas con la aptitud biológica. Si X_A es la frecuencia de la población A y X_B es la frecuencia de la población de B, entonces el costo esperado para la población A y B respectivamente está dado por:

$$\begin{aligned}f_A &= ax_A + bx_B \\f_B &= cx_A + dx_B\end{aligned}$$

Estas ecuaciones asumen que para cada jugador, la probabilidad para interaccionar con un jugador A es x_A y para interaccionar con un jugador B es x_B . En general la dinámica evolutiva de dos estrategias (A y B) genera 4 resultados posibles (Figura 9). Uno es la dominancia (Figura 9a) ya sea de A o de B, esto significa que la selección llevará a un estado donde la población total estará compuesta de jugadores que emplean la estrategia A (es decir $x_A = 1$) o de jugadores que sólo emplean la estrategia B ($x_B = 1$).

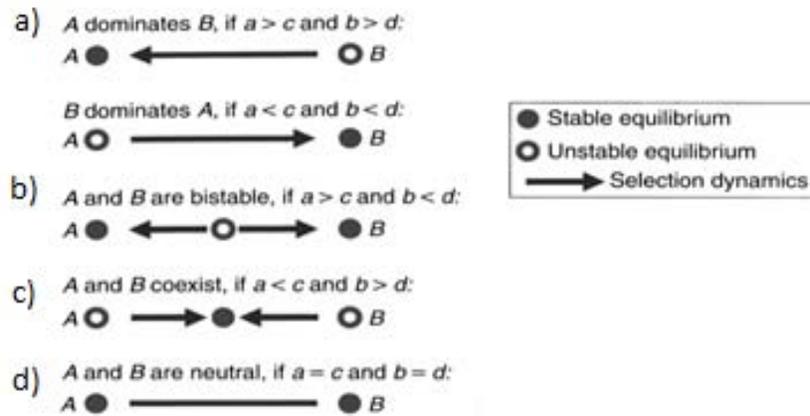


Figura 9. Dinámica evolutiva de dos estrategias que arrojan 4 resultados posibles. Modificada de Nowak, 2006.

Otro resultado puede ser la biestabilidad (Figura 9b) donde la estrategia A es la mejor respuesta contra la misma estrategia A, al igual que B y para la dinámica de selección dentro de la población, este resultado depende de las condiciones iniciales. También el resultado puede ser la coexistencia (Figura 8c) de ambas estrategias, esto es que la estrategia A es la mejor respuesta contra la estrategia B y viceversa, así que una población donde coexisten A y B convergerá a un equilibrio interior estable. Y por último la neutralidad (Figura 9d) donde las frecuencias de A y B solo son sometidas por la deriva génica ya que no importa qué estrategia se utilice se obtendrá la misma recompensa, de tal forma que la selección no cambiará la composición de la población.

Un caso interesante es cuando existen 3 estrategias ya que los resultados pueden ser considerablemente más complicados debido a que se pueden generar comportamientos cíclicos del tipo de piedra, papel y tijeras. En este juego hay 3 estrategias las cuales se reprimen o dominan (como se quiera ver) mutuamente de forma cíclica. La estrategia B domina a la estrategia A, la estrategia C domina a B y A, en cambio, domina a C completando el ciclo (Figura 10a). Así la dinámica en este juego puede llevar, dependiendo de la fuerza relativa de las interacciones, ya sea a una coexistencia a largo plazo de las tres estrategias o bien a un aumento cada vez mayor en las oscilaciones, finalizando eventualmente, con la eliminación de dos de las estrategias (Nowak y Sigmund, 2004).

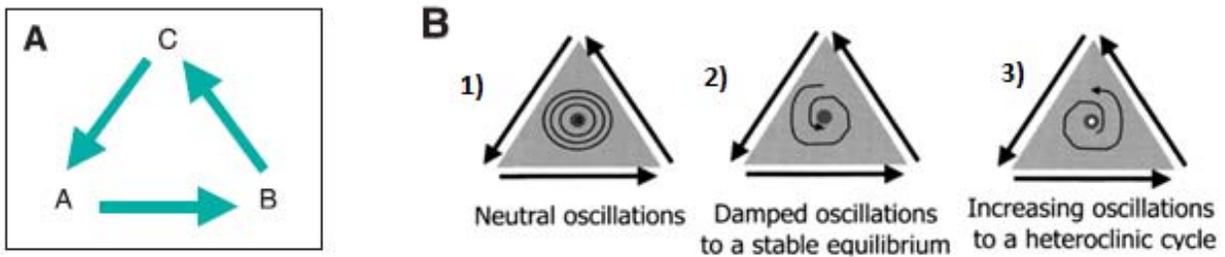


Figura 10. A) Dinámica dominante en tres estrategias del tipo piedra, papel y tijeras. B) espacio fásico de una estrategia 3 x 3 y sus diferentes resultados.

La matriz de recompensa nos da 2 posibilidades: 1) si la determinante de A es positiva (que es el caso para $a_1a_2a_3 < b_1b_2b_3$), entonces existe un equilibrio interior único que es globalmente estable. Las trayectorias de la dinámica replicadora empezando desde cualquier condición interior inicial convergerán a este equilibrio en una forma oscilatoria (Figura 10B-1). Si la determinante de A es negativa (que es el caso para $a_1a_2a_3 > b_1b_2b_3$), entonces existe un equilibrio interior único que es inestable. Las trayectorias de la dinámica replicadora que empiezan en cualquier condición inicial interior convergerán en los límites del simplex en oscilaciones con aumento en su amplitud. El límite del simplex (2-simplex de 3 nodos) es un ciclo heterocíclico que es un atractor para todas las trayectorias de partida en el interior (Figura 10B-3) (Nowak, 2006).

Von Neuman y Morgersten se interesaron mucho en los juegos de suma cero donde se asume que los jugadores maximizarán su desempeño de una manera completamente racional, sin embargo en la teoría de juegos evolutiva este principio de racionalidad falla completamente. La lucha por la supervivencia evolutiva es acerca del éxito de la reproducción individual, sin embargo la selección natural (en algunos casos) lleva al desarrollo de la cooperación natural de ciertas comunidades de animales y otros seres vivos. Una respuesta a esto vino de la teoría de los autómatas celulares originada por Von Neumann en la década de los 40's, cuyo idea era crear robots autoreproducibles por lo que tenía que crear un mundo artificial cuyas leyes naturales fueran completamente conocidas para nosotros. La conexión entre este mundo artificial y la teoría de juegos se dio gracias al trabajo del teórico evolutivo Martin Novak. El analizó una población hipotética que vivía un tablero de ajedrez: cada individuo se encontraba en cada cuadrado del tablero jugando el dilema del prisionero con cada uno de sus 8 vecinos y la estrategia de cada individuo está predeterminada. La estrategia dicta que o solamente coopera o se rehúsa a cooperar por completo

Capítulo 3

Modelación basada en agentes

En este capítulo se describen las herramientas computacionales para el diseño de modelos biológicos y los conceptos para entender los modelos basados en agentes. Se detallará especialmente el programa NetLogo el cual es un ambiente de simulación multiagente ampliamente utilizado, tanto por estudiantes como por académicos en el campo social, computacional y en las ciencias exactas. Se describirán algunas aplicaciones de los modelos basados en agentes en el campo de la biología y su impacto para la generación de nuevos métodos en la investigación (Railsback y Grimm, 2009; Macal y North, 2010; Wilensky, 1999).

3.1 Introducción a la modelación basada en agentes

Un modelo es una representación intencionada de algún sistema real. Los modelos son creados y utilizados para resolver problemas o responder preguntas acerca de un sistema o varias clases de sistemas. Generalmente en ciencia lo que se quiere es entender cómo funcionan las cosas, explicar ciertos patrones que son observados, y predecir el comportamiento de un sistema en respuesta a un cambio. Dado que los sistemas reales, por lo general, son demasiado complejos o su desarrollo es muy lento para ser analizado experimentalmente, hay esfuerzos por formular una representación simplificada del sistema utilizando ecuaciones o programas computacionales con los cuales podemos experimentar y manipular los datos (Railsback y Grimm, 2009). Los acercamientos matemáticos y computacionales tradicionales al entendimiento de los sistemas biológicos se han enfocado en la modelación de poblaciones homogéneas de moléculas, células u organismos y cómo cambian a través del tiempo. Estos modelos se han concentrado en las dinámicas intracelulares, por ejemplo, cómo las concentraciones de químicos esenciales, mRNA y proteínas varían en un lapso de tiempo determinado dentro de la célula. Aunque este acercamiento es esencial para entender las acciones de las células individuales, no nos permite entender el comportamiento que ocurre a nivel poblacional debido a las interacciones entre células individuales de la población. Así que un método común para capturar las interacciones celulares que ocurren a una escala menor y que dan lugar a diferentes comportamientos en una escala poblacional, es empleando modelos basados en agentes (ABMS) o modelos basados en individuos (IBMS). Estos modelos permiten un entendimiento de la relación entre las reglas microscópicas de los agentes y los comportamientos macroscópicos de la población (Gorochowski et al., 2012).

Entonces los ABMS son modelos donde los individuos u agentes son descritos como entidades únicas y autónomas las cuales, usualmente, tienen interacciones locales. Los agentes pueden ser organismos, personas, empresas, instituciones o cualquier otra entidad que siga cierto objetivo. Ser único implica que los agentes usualmente son diferentes unos de otros en características tales como el tamaño, su ubicación, reservas de recursos, su historia, etc. Interacciones locales significa que los agentes, usualmente, no interaccionan con todos los demás agentes sino solo con sus vecinos más cercanos (en un espacio geográfico o algún otro tipo de espacio como una red). Ser autónomo implica que los agentes actúan independientemente unos de otros y persiguen sus propios objetivos. Así los agentes muestran un comportamiento adaptativo ya que estos ajustan su comportamiento dependiendo del estado actual de cada uno, de otros agentes y de su ambiente.

Los modelos basados en agentes tienen sus raíces históricas directamente con los sistemas adaptativos complejos (CAS) y bajo la noción de que los sistemas se construyen desde sus cimientos hacia arriba, en contraste con el punto de vista de que los sistemas deben ser diseñados desde arriba hacia abajo como en la dinámica de sistemas. Si bien en el comienzo se modelaban fenómenos biológicos (principalmente en ecología) desde una perspectiva individual, ya se empleaba esta noción de analizar los sistemas desde sus componentes hasta las estructuras y propiedades emergentes que generaban (Lorek et al., 1999). Los sistemas adaptativos complejos se ocupan del problema de cómo los comportamientos complejos surgen en la naturaleza a partir de agentes autónomos “miopes”. Así el campo de los CAS estuvo originalmente motivado por las investigaciones sobre la adaptación y la emergencia en sistemas biológicos. John Holland identificó propiedades y mecanismos comunes a todos los CAS. Entre las propiedades se encontraban: 1) la agregación: que permitía la formación de grupos; 2) la no linealidad: invalida la simple extrapolación; 3) flujo: permite la transferencia y transformación de recursos e información; 4) diversidad: permite a los agentes comportarse de forma diferente uno de otro y que con frecuencia lleva a la robustez del sistema. Por otro lado los mecanismos de los CAS son: 1) etiquetado: que permite a los agentes ser nombrados y reconocidos; 2) modelo interno: que permite a los agentes razonar sobre su mundo externo; 3) bloques de construcción: que permite a los componentes y a todo el sistema descomponerlos en componentes más simples. Estas propiedades y mecanismos de los CAS proveyeron de un marco de trabajo útil para el diseño de la modelación basada en agentes. Posteriormente estas ideas fueron plasmadas en otra teoría matemática igualmente importante que ilustró las ideas básicas de la modelación basada en agentes, el juego de la vida creado por John Conway en 1970 (Macal & North, 2010). El juego de la vida está basado en los autómatas celulares (CA) y como se describió en el capítulo 2 a partir de reglas simples la interacción de varias celdas exhibe comportamientos emergentes. Thomas Schelling (Fig 11) se le acredita por desarrollar la primera simulación de un modelo social basado en agentes, donde los agentes representan las personas y las interacciones entre agentes

representan un proceso social relevante. Schelling aplicó las nociones de los autómatas celulares para estudiar los patrones de segregación de las casas con diferentes o iguales coloraciones. Así el modelo de Schelling mostró que ciertos patrones pueden emerger sin necesariamente tener implicados o ser consistentes con los objetivos de los agentes individuales. Esto fue una importante observación que estimuló el interés y le dio dirección al campo de los ABMS (Macal & North, 2005).

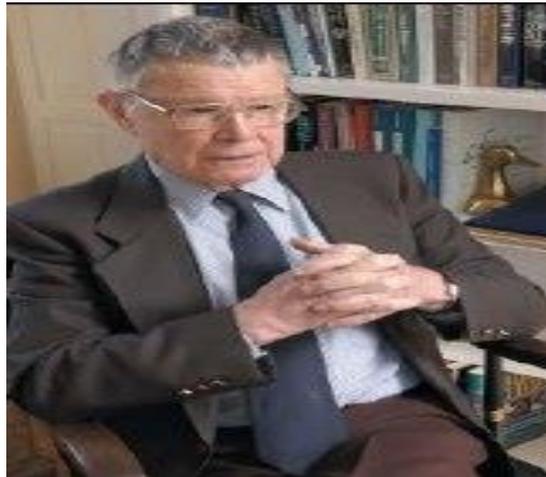


Figura 11. Fotografía de Thomas Schelling, premio nobel de economía en el 2005 e importante investigador en el campo de la economía computacional basada en agentes.

Las aplicaciones de los modelos basados en agentes abarcan un amplio rango de áreas y disciplinas. En la economía es aplicable siguiendo la teoría estándar de la microeconomía lo que ha creado el campo de la economía computacional basada en agentes. En la antropología también se emplea para desarrollar simulaciones basadas en agentes, a gran escala, de antiguas civilizaciones que ayuden a explicar su crecimiento y decaimiento basados en datos arqueológicos. También la modelación basada en agentes se ha utilizado en la sociología, para modelar la vida social de cómo influyen las personas unas con otras y en la política para entender los procesos básicos involucrados en la identidad nacional. En las ciencias como la física y la biología también han sido empleados como complemento a la investigación teórica y experimental; en física principalmente se emplea en la modelación de la emergencia de posibles estructuras como resultado del autoensamblaje molecular y en la biología en la modelación del comportamiento y auto organización de colonias bacterianas así como la interacción a múltiples escalas de los diferentes niveles de organización (Macal & North, 2005).

La complejidad en los modelos a menudo no se limita por el propósito del modelo sino por la tratabilidad matemática del mismo. Cuando el cálculo diferencial era el único acercamiento que se tenía para modelar, los modelos debían ser lo más simple posibles para resolverse de forma

matemática, por lo que la modelación estaba limitada a problemas simples. Sin embargo con la simulación computacional, las limitaciones de la tratabilidad matemática desaparecen, por lo que es posible abordar problemas que requieren menos simplificación e incluir más características del sistema real (Railsback y Grimm, 2009). El desarrollo de programas computacionales dedicados a lo ABMS aún sigue teniendo ciertos obstáculos sobre todo porque los investigadores dedicados a la enseñanza en diferentes campos de la biología, economía, sociología y política no están capacitados para transmitir estas herramientas a sus estudiantes. Sin embargo existe una gran variedad de plataformas computacionales, las cuales se han vuelto bastante populares en los últimos años ya que proveen diseños estandarizados y herramientas que no limitan el tipo de complejidad que pueden implementar y que permite generar una librería de diferentes marcos de modelos con varias aplicaciones. El uso de uno u otro depende, en general, en su dificultad de uso: que tenga las suficientes herramientas para construir el modelo deseado, que tenga buena capacidad para observar y ejecutar simulaciones experimentales y que tenga buena disposición de documentación y comunicación del programa. Algunos de los software más empleados son: Java Swarm, Repast, Netlogo y MASON (Tabla 2). Netlogo resulta bastante adecuado para la modelación de sistemas que no son extremadamente complejos donde los agentes tienen interacciones locales y en tiempos cortos y por la sencillez y fácil uso que permite diseñar modelos rápidamente.

	Term			
Concept	MASON	NetLogo	Repast	Swarm
Object that builds and controls simulation objects	Model	Observer	Model	Modelswarm
Object that builds and controls screen graphics	ModelWithUI	Interface	(None)	Observerswarm
Object that represents spce and agent locations	Field	World	Space	Space
Graphical Display of spatial information	Portrayal	View	Display	Display
User-opened display of an agent's state	Inspector	Monitor	Probe	Probedisplay
An agent behavior or event to be executed	Steppable	Procedure	Action	Action
Queue of events executed repeatedly	Schedule	Forever procedure	Schedule	Schedule

Tabla 2. Algunos de los softwares más utilizados para la modelación basada en agentes difieren en algunos de sus conceptos para el diseño de los modelos.

3.2 Netlogo

Netlogo es un entorno de modelado multi-agente programable. La parte de Logo, es porque NetLogo forma parte del dialecto del lenguaje de la familia Logo. La parte de Net, trata de evocar la naturaleza descentralizada e interconectada de los fenómenos que pueden ser modelados con el programa y a su vez también hace referencia a HubNet, el ambiente de simulación multiusuario participativo incluido en Netlogo (Wilensky, 1999). Fue lanzado en 1999 y puede correr en la mayoría de las plataformas computacionales actuales (Microsoft Windows, Mac OS X y Linux) y la versión más reciente actualmente es la 5.1.0. Netlogo está escrito en Java lo que permite a los programadores salvar sus modelos en applets y publicar su simulación construida en Netlogo, directamente en una página web. Netlogo emplea una interfaz gráfica, la cual contiene tres pestañas: ejecutar, información y código (Sklar, 2007).

La pestaña de ejecutar es esencialmente un editor visual en el cual el programador puede crear y editar elementos gráficos: botones (que llaman a un procedimiento), deslizadores (permiten cambiar los valores numéricos de cierta variable en un rango específico), interruptor (permite activar o desactivar un valor numérico de una variable), seleccionador (permite escoger un valor de una variable establecida por el programador), monitor (muestra el valor de una variable mientras se corre la simulación), gráfico (muestra las gráficas de ciertas variables mientras corre la simulación), entre otras (Fig. 12). El contenido de estos elementos es controlado por variables globales y métodos, los cuales son especificados en la sección del código (Sklar, 2007).

Por default se presenta una pantalla llamada mundo, donde corre la simulación y cuyo tamaño es de 35X35 unidades o patches, pero puede ser cambiada en la sección de configuración. También puede cambiarse el tamaño de los patches y el origen del mundo que por default se encuentra en el centro, por lo que las coordenadas de los bordes (en patches) se extienden desde (-17, 17) de la esquina superior izquierda, hasta (17, -17) en la esquina inferior derecha.

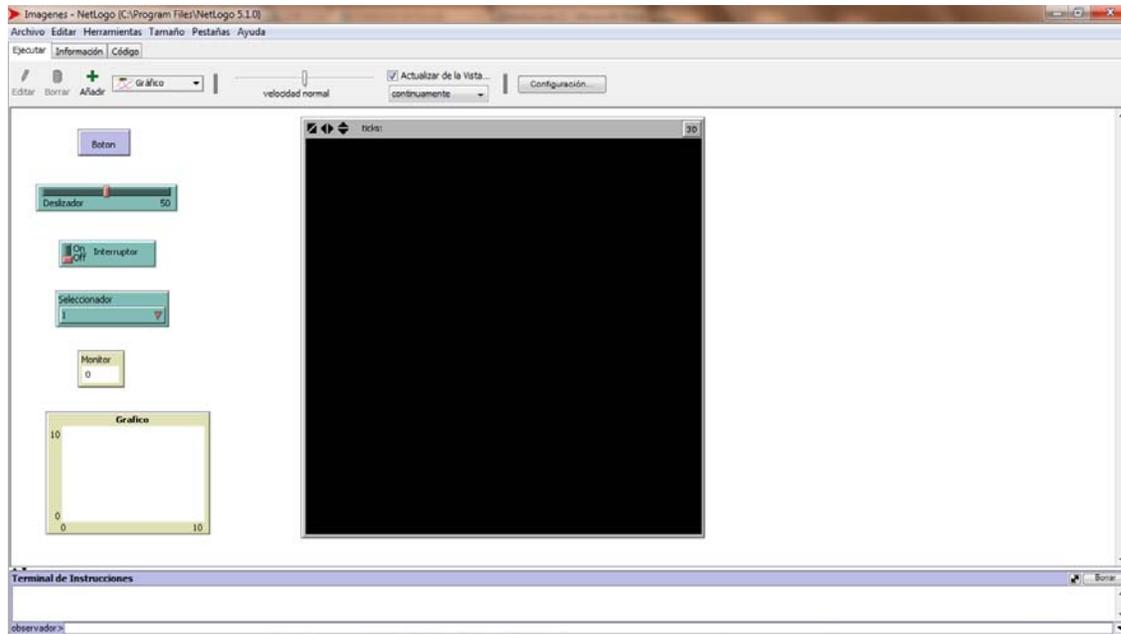


Figura 12. Imagen del entorno de Netlogo donde se muestran algunos de los componentes presentes en la pestaña de ejecutar.

Las unidades básicas de simulación en Netlogo son las tortugas, que corresponden a los agentes. Cada tortuga reside en un patch y múltiples tortugas pueden localizarse en el mismo patch. Todos los movimientos de las tortugas están computarizados en términos de patches, así por ejemplo, el comando `forward 4` mueve a las tortugas hacia adelante 4 patches sobre 4 tiempos en la simulación. Así la idea principal es que las tortugas representan las entidades físicas cuyo comportamiento resulta en el movimiento a través del mundo de dos dimensiones. Sin embargo este no es un requerimiento para todas las simulaciones, algunos modelos en la biblioteca de NetLogo y otros más disponibles en varias páginas web ofrecen otras formas de emplear las tortugas según el fin para el que fue elaborado el modelo en diferentes áreas de estudio. Además se puede modificar la forma, el tamaño y el color de cada una de las tortugas e incluso diseñar las propias, todo esto con la herramienta de editor de tortugas (Sklar, 2007).

La pestaña de información está diseñada para permitir y alentar la documentación del modelo de NetLogo. El programador con el botón de editar puede introducir el contenido de la información en formato de texto lo cual permite una mejor presentación de la documentación del modelo, siendo más fácil de entender en qué consiste y para que puede resultar útil el modelo (Fig. 13).

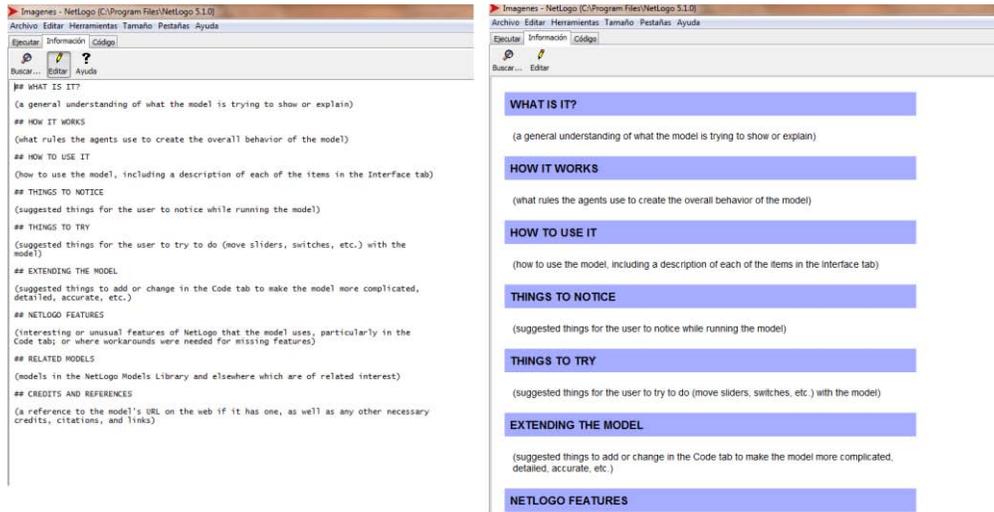


Figura 13. Imagen de la pestaña de información de NetLogo donde es posible documentar nuestro modelo de forma fácil y rápida.

La pestaña de código permite al programador hacer uso de un lenguaje bastante extenso de primitivas que controlan el comportamiento de los elementos y tortugas, en la parte de ejecutar de NetLogo. Una primitiva es uno de los procedimientos o comandos incorporados en NetLogo que les dice a los agentes que debe hacer. Estas primitivas son sumamente importantes ya que son básicamente quienes realizan el trabajo computacional del programa. Las primitivas pueden agruparse de acuerdo a su funcionalidad. Algunas operan a nivel de las estructuras de datos (turtles, agentset, list y string). Algunas controlan el flujo básico del programa y su funcionalidad. Otros proveen acceso a los componentes de la interface (ejemplo, patch y world) (Sklar, 2007). En general NetLogo tiene 2 tipos principales de primitivas: comandos y reporteros (Fig 14). Los comandos, simplemente le dicen al agente que haga algo (por ejemplo “move-to”). Los reporteros calculan un valor y lo reportan de vuelta para emplearlo en el programa (por ejemplo “random-float”). Todas estas primitivas se encuentran definidas en el diccionario de NetLogo al cual se puede acceder directamente desde el programa en la parte de ayuda. Por otro lado el término “contexto” en Netlogo es muy importante ya que cada pieza del código, en el programa, esta “en el contexto” de algún (u ocasionalmente varios) tipo de agente: turtles, patches, links o el mismo observador; y solo puede ser ejecutado por ese tipo de agente (Railsback y Grimm, 2009). Netlogo ha sido ampliamente utilizado en el desarrollo de modelos para estudiar la emergencia de patrones así como de diferentes fenómenos que ocurren en la naturaleza (Wilensky, 2002).

```
Bounce Example - NetLogo
Archivo Editar Herramientas Tamaño Pestaaas Ayuda
Ejecutar Información Código
Borrar... Comprobar Procedimientos Síngnado automático

to setup
  clear-all
  set-default-shape turtles "circle"
  draw-walls
  crt 1 ; create some turtles
  [ randomize ] ; place them randomly
  reset-ticks
end

; draw the boundaries (walls) of the "billiard table"
to draw-walls
  ; draw left and right walls
  ask patches with [abs pxcor = max-pxcor]
  [ set-pcolor blue ]
  ; draw top and bottom walls
  ask patches with [abs pycor = max-pycor]
  [ set-pcolor blue ]
end

; set random location
to randomize
  setxy random-pxcor random-pycor
  if pcolor = blue ; if it's on the wall...
  [ randomize ] ; ...try again
end

to go
  ask turtles [
    ifelse leave-trace? ; the turtle puts its pen up or down depending on the
      [ pu ] ; value of the LEAVE-TRACE switch
      [ pd ]
    bounce
    fd 0.1
  ]
  tick
end

; this procedure checks the coordinates and makes the turtles
; reflect according to the law that the angle of reflection is
; equal to the angle of incidence
to bounce ; turtle procedure
  ; check: hitting left or right wall?
  if abs [pxcor] of patch-ahead 0.1 = max-pxcor
  [ set-heading (- heading) ]
  ; check: hitting top or bottom wall?
  if abs [pycor] of patch-ahead 0.1 = max-pycor
  [ set-heading (180 - heading) ]
end
```

Figura 14. Ejemplo de algunos comandos y reporteros empleados en NetLogo para el diseño de un modelo (código tomado de la biblioteca de modelos).

Parte II

Metodología

4. Hipótesis

Si la regulación transcripcional de la construcción genética elaborada por el equipo de Elowitz sigue un comportamiento cíclico periódico parecido al juego de piedra papel y tijeras y esta dinámica se encuentra acoplada a un sistema de comunicación por quórum sensing será posible estabilizar los patrones globales de expresión, para ser sincronizados en la población de bacterias de manera global, al modular la densidad celular así como la difusión intra y extracelular del autoinductor.

5. Objetivos Generales

Obtener un modelo computacional basado en agentes que ayude a entender y a simular el proceso de sincronización molecular en una población bacteriana. Modelar computacionalmente la interacción entre células por medio de un sistema de difusión llamado percepción por quórum y determinar si modificando la producción de esta molécula se genera una sincronización de las células.

5.1 Objetivos particulares:

- Diseñar la regla de la dinámica del sistema empleando la teoría de juegos así como el modelo matemático propuesto por Elowitz *et al.*
- Desarrollar el código en Netlogo para construir un modelo espacial que simule la interacción de los componentes del sistema poblacional bacteriano con la construcción genética desarrollada por Elowitz y colaboradores.
- Analizar los resultados de las simulaciones obtenidas y proponer una mejora para la implementación experimental.

6. Material y método

Para poder desarrollar el modelo computacional primero se revisó el modelo de ecuaciones diferenciales propuesto por Elowitz y colaboradores, especialmente la parte de la incorporación del sistema de quórum sensing al modelo del represilador y a su vez incluir la matriz de pagos que determinaría los costos de emplear cada una de las estrategias génicas (este concepto sintetiza los procesos de transcripción de los genes y producción de proteína). Posteriormente se llevaron a cabo prácticas con base en el manual que ofrece el programa Netlogo así como del libro de Raisback y colaboradores y de distintos tutoriales que ofrece la red (como las lecciones ofrecidas en el curso online de complejidad de la página complexity explorer de la Universidad de Santa Fe). Después de haber adquirido conocimiento y habilidad en el manejo del programa se desarrolló el modelo siguiendo el protocolo ODD, el cual es un protocolo ampliamente utilizado por diversas áreas de estudio para desarrollar modelos basados en agentes (Raisback y Grimm, 2009). Y por último se llevaron a cabo las simulaciones en el programa Netlogo con parámetros de las variables establecidos según el modelo matemático. Por lo anterior este trabajo se desarrolló en las siguientes etapas:

1. Revisión del modelo matemático y la matriz de pagos
2. Aprendizaje del uso del software
3. Desarrollo del código en Netlogo empleando el protocolo ODD
4. Llevar a cabo las simulaciones

6.1 Revisión del modelo matemático y la matriz de pagos

El modelo matemático propuesto en el artículo de Elowitz consiste en un conjunto de ecuaciones diferenciales las cuales describen el comportamiento del sistema molecular. Un conjunto de 4 ecuaciones describe la dinámica de la expresión genética en la población celular (dinámica del mRNA) así como la producción de las proteínas correspondientes a cada uno de los genes del represilador. Este modelo matemático se utilizó como herramienta para determinar la actividad del autoinductor en el modelo de Netlogo. Las 3 ecuaciones que gobiernan la dinámica del represilador son las siguientes:

$$\begin{aligned}\frac{da_i}{dt} &= -a_i + \frac{\alpha}{1 + C_i^n} \\ \frac{db_i}{dt} &= -b_i + \frac{\alpha}{1 + A_i^n} \\ \frac{dc_i}{dt} &= -c_i + \frac{\alpha}{1 + B_i^n} + \frac{kS_i}{1 + S_i}\end{aligned}$$

donde a_i , b_i y c_i son las concentraciones de mRNA de *tetR*, *cl* y *lacI* respectivamente de cada una de las células. La concentración de las proteínas está denotado por A_i , B_i y C_i y la concentración del autoinductor (AI) intracelular se denota por S_i . El modelo es adimensional ya que mide el tiempo con respecto a la tasa de vida media del mRNA y los niveles de las proteínas en unidades de sus constantes de Michaelis. La concentración de S_i también se mide con base a su constante de Michaelis. El término α corresponde a la tasa de transcripción en ausencia de represor y también es adimensional. Por otro lado el termino k se refiere a la contribución del autoinductor en la transcripción del gen *cl* y el termino n es el coeficiente de Hill, que se refiere a la cierta cooperatividad en la actividad represora (Garcia-Ojalvo *et al.*, 2004).

El último término de la tercera ecuación, fue el término empleado para incorporar la dinámica del quórum sensing a la del represilador. Dado que la matriz de pagos obtenida de la teoría de juegos (de la que se hablara a continuación) establece el comportamiento cíclico oscilatorio de los genes, este último término de la ecuación se emplea para acoplar dicho comportamiento con la comunicación celular mediada por la percepción de quórum. Las ecuaciones que gobiernan la difusión del autoinductor dentro y fuera de la célula son las siguientes:

$$\frac{dS_i}{dt} = -k_{s_0}S_i + k_{s_1}A_i - n(S_i - S_e)$$

donde S_i es la concentración intracelular y S_e la concentración extracelular del autoinductor. Se asumen las tasas de vida iguales tanto para *TetR* como para *LuxI* por lo que se usa la misma variable (A_i) en la ecuación. K_{s0} y K_{si} corresponden a constantes de degradación y síntesis y el termino n corresponde a la tasa de difusión a través de la membrana de las células. Por otro lado:

$$\frac{dS_e}{dt} = -k_{se}S_e + n_{ext} \sum_{j=1}^N (S_j - S_e)$$

o bien,

$$\frac{dS_e}{dt} = -k_{se}S_e + k_{diff}(\bar{S} - S_e).$$

Donde k_{se} es la constante de degradación y n_{ext} y k_{diff} son la tasa de difusión del autoinductor. Con estas 2 ecuaciones se implementó el transporte del autoinductor dentro y fuera de las células en el modelo de Netlogo.

Existen varios ejemplos conocidos en ecología donde se aplica el juego de piedra papel y tijeras para describir el comportamiento de poblaciones de especies, donde la selección dependiente de la frecuencia juega un papel esencial en el comportamiento de la población (Sinervo & Lively, 1996; Frean & Abraham, 2001). El costo de emplear estas estrategias en una dinámica dominante cíclica puede establecerse por medio de una matriz de pagos 3 X 3. Si se trata de una matriz simétrica, es posible transformar esta matriz de pagos en una matriz de pagos que contiene entradas con valor cero en su diagonal, si se sustrae el elemento diagonal de cada una de las columnas. Por ejemplo en la siguiente matriz (Figura 15a) se sustrae los valores 4, 1 y 2 de la diagonal y queda una matriz de valor cero en la misma.

a)

$$A = \begin{matrix} & \begin{matrix} R & S & P \end{matrix} \\ \begin{matrix} R \\ S \\ P \end{matrix} & \begin{pmatrix} 4 & 2 & 1 \\ 3 & 1 & 3 \\ 5 & 0 & 2 \end{pmatrix} \end{matrix}$$

4 1 2

Se sustrae a cada elemento horizontal el valor de cada elemento de la diagonal

b)

$$A = \begin{matrix} & \begin{matrix} R & S & P \end{matrix} \\ \begin{matrix} R \\ S \\ P \end{matrix} & \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} \end{matrix}$$

Figura 15. Matriz de pagos 3 x 3. Después de sustraer los valores de la diagonal a los valores de las filas de la matriz queda una matriz con valores en la diagonal iguales a cero.

Esta matriz de pagos describe un juego de suma cero. Para los juegos de suma de cero tenemos que

$$a_{ij} = -a_{ji}.$$

Por lo que la ganancia de uno de los jugadores es la pérdida del otro, es decir que la adecuación de la población es siempre cero. Empleando los valores de la matriz de pagos de la Figura 15b. como el costo por emplear cada una de las estrategias (genes) del represilador, se desarrolló la regla de transición de los agentes (células) al interactuar unas con otras en Netlogo.

6.2 Aprendizaje del uso del software

El primer paso en el aprendizaje del uso de este software fue la lectura del manual de Netlogo que viene incluido en el programa o que también puede consultarse en la página <https://ccl.northwestern.edu/netlogo/docs/> para familiarizarse con los términos empleados en el desarrollo del código, así como de la interfaz donde se corre la simulación. Posteriormente el libro de Railsback y colaboradores (200) valió una fuente de ejemplos y modelos a desarrollar para poner en práctica los conocimientos adquiridos. Igualmente tutoriales en forma de videos y en diversas paginas autónomas (revisar referencias de las páginas web). Fueron consultados a través de la red y en especial la página <http://stackoverflow.com/> fue de gran ayuda para realizar consultas en una comunidad de programadores y personas involucradas en el desarrollo de modelos computacionales, ya que este sitio permite discutir, preguntar y compartir el conocimiento sobre el uso de estos programas.

6.3 Protocolo ODD

Los modelos basados en agentes fueron al principio de su desarrollo criticados por ser pobremente documentados, por lo que estos modelos no podían ser evaluados adecuadamente por la comunidad científica. Estas críticas motivaron el desarrollo del protocolo ODD, el cual pretende crear un formato genérico y una estructura estándar con la cual todos los modelos basados en agentes puedan ser documentados (Grimm, et al., 2010). El acrónimo ODD se establece por los 3 bloques principales de información que se describen en el protocolo: overview (visión general), design concepts (diseño de conceptos) y details (detalles). Esencialmente el protocolo ODD es una

lista de decisiones que un modelador debe hacer para crear un modelo basado en agentes que pueda ser implementado y analizado (Railsback & Grim, 2009). De tal forma que el propósito principal del protocolo ODD es hacer las descripciones del modelo lo más fácil y eficiente posible y que lleve, incluso, a una descripción más completa del modelo. Esto hará al modelo basado en agentes más fácil de replicar y por lo tanto no será menospreciado por no tener carácter científico. Como se mencionó anteriormente el protocolo ODD se sustenta en 3 bloques principales, estos bloques a su vez contienen ciertos elementos que deben ser explicados para describir correctamente el modelo. Con base en estos elementos, el modelo de Netlogo se desarrolla de la siguiente manera.

6.3.1 Visión general y planificación

Propósito

El propósito principal de este modelo es explorar el efecto del acoplamiento mediado por quórum sensing en una colonia bacteriana que tiene incorporada la construcción genética “represilador” elaborada por el equipo de Elowitz. Se pretende observar cómo los agentes (las bacterias, estrategias génicas y autoinductores) interactúan unos con otros y como esta interacción modifica la expresión genética de los genes en el sistema del represilador, de tal forma que la transcripción de los genes dentro de cada célula individual se regule generando un comportamiento de acoplamiento emergente en toda la población.

Entidades, variables de estado y escalas

Las entidades en el modelo son: las células bacterianas, las moléculas de quórum sensing (autoinductores) así como las estrategias génicas, todos estos corresponden a las tortugas. Las estrategias génicas en sí son la representación de varios procesos moleculares como son la transcripción de los genes y la producción de proteínas. Dado que la actividad cíclica típica del represilador comprende en realidad la interacción de 2 componentes en diferentes niveles moleculares (gene- proteína) esto no correspondería a la interacción que se da en el programa de agente con agente, por lo que se simplifica este proceso como de interacción de estrategias representadas por los agentes. El ambiente en donde habitan o el mundo es representado por los patches y el observador es quien manipula todas estas entidades. Todas las tortugas tienen diferentes estados de localidad: las estrategias génicas se encuentran restringidos en 6 diferentes celdas en el caso del modelo poblacional o a 3 celdas que corresponden a las 3 diferentes células bacterianas (en el caso del modelo de 3 líneas celulares). Las células bacterianas igualmente se encuentran distribuidas en una celda mayor de forma aleatoria. Las moléculas de quórum sensing

aparecen también en la celda de las células bacterianas pero tanto unas como las otras pueden pasar a las celdas de las estrategias génicas. Las estrategias agencias a su vez tienen 3 estados de coloración: verde, rojo y azul. Las bacterias tienen dos estados de coloración: gris y amarillo. Las células tienen 1 estado cuantificable llamado autoinducción que muestra el nivel de autoinductores que ha absorbido la célula. Los autoinductores tienen el estado de coloración amarillo y su posición es igualmente aleatoria. El medio donde habitan (mundo) es una parcela de patches de 40 x 40. Los patches que tienen la función de delimitar las celdas tendrán el estado de coloración blanco formando una línea. El observador creará otras entidades que pertenecen a la parte de la interfaz del programa como son: los botones de gráfica para cuantificar la población de las estrategias génicas, el autoinductor y la autoinducción; sliders para definir la población de las estrategias génicas y otras constantes, etc. Por otra parte la escala temporal se mide por medio de pasos discretos llamado ticks que el programa utiliza para cuantificar la simulación del código. La cantidad de ticks empleados puede depender del tipo de simulación que se pretenda realizar.

Panorama general de los procesos y planificación

La idea general del desarrollo de los procesos es, crear primero, los agentes y delimitar el mundo a las regiones correspondientes donde se establecen dichos agentes. Posteriormente vienen los procesos necesarios para general la movilidad de los agentes y las reglas de interacción entre ellos. Y finalmente los procesos que permiten la visualización de la información generada a partir de los procesos anteriores; no tienen un orden en sí, ya que estos procesos no están ligados a las entidades del modelo, siendo procesos que están enfocados para el observador, es decir, se especifican procesos para mostrar los diferentes estatus del modelo en representaciones gráficas o resúmenes estadísticos (Grimm et al, 2010).

La primer parte del modelo corresponde a la creación de los agentes. Se deben establecer los criterios que tendrán cada uno de los agentes como son sus variables globales (colores) o variables propias (inducción) y también definir cada tipo de agentes de acuerdo al papel y variables que juegan dentro del mundo (breeds). Después es necesario describir los procedimientos para la creación de los agentes y del mundo de acuerdo a sus variables de estado y comportamiento (tamaño, color, coordenadas, forma, etc.), y su distribución (por ejemplo los juegos de genes establecidos en 6 secciones diferentes). Así también la relación con la parte de la interfaz (botones, sliders) que afecte la cantidad de agentes que deberán ser creados (células y genes).

La segunda parte corresponde a los procesos que hacen que cambien los estados de las variables de las entidades del modelo. Estos procesos pueden subdividirse en 4 categorías: los procesos relacionados con el movimiento de los agentes (bounce and move); los procesos relacionados con la teoría de juegos y el represilador de Elowitz (play); los procesos relacionados con la difusión del autoinductor dentro y fuera de la célula (difusión-in, difusión-out y lac-recali); y los procesos relacionados con la fluorescencia (fluorescence). El movimiento de los agentes debe darse aleatoriamente y el rango del desplazamiento debe ser de 1 patche. En el caso del procedimiento bounce, además del movimiento típico anterior, los agentes deben delimitar su movimiento al encontrarse una barrera (formada por una línea al cambiar el color de los patches con un límite en las coordenadas del eje x y a lo largo del eje y) lo que ocasione un cambio contrario de la dirección a la que se dirigían. La acción del represilador constituye 3 agentes (estrategias génicas) que se reprimen cíclicamente como en el juego de piedra papel y tijeras, de tal forma que cambien su color (dado por la variable global colors) siguiendo las reglas del juego de piedra papel y tijeras, lo que permita una dinámica poblacional oscilatoria. Esto dependiendo de la composición con que esté la población de agentes según las estrategias o colores presentes. Esta interacción no se presenta directamente en el sistema biológico real ya que en realidad se da entre proteínas y regiones promotoras, sin embargo el comportamiento que genera el sistema es el mismo que generaría el juego de piedra papel y tijeras, así se hace la suposición de que el juego se da entre interacción directa de los genes. La difusión del autoinductor está definido por las ecuaciones del modelo de Elowitz (Garcia-Ojalvo et al., 2004) aplicadas a los agentes (células, genes y autoinductor). El procedimiento de inducción (difusión-in) debe ser realizado por las células y de forma aleatoria. El procedimiento de creación de autoinductores (difusión-out) debe ser realizado por los patches ya que estos se crean en el mundo y su creación no depende de los autoinductores (refiriéndose a los agentes) presentes anteriormente. La acción de la difusión dentro y fuera de la célula también afecta a la población de los genes del anterior procedimiento, por lo que la ecuación del modelo de Elowitz también se emplea en esta parte para redefinir la composición de la población de cada uno de los juegos de estrategias génicas (lac-recali). Las constantes y parámetros utilizados también se definen aquí y se le asigna su relación con los botones o sliders de la interfaz para definir el valor que empleara el código que contiene las ecuaciones siguiendo los procedimientos correspondientes. Por último la fluorescencia toma la variable de la inducción para establecer el cambio de color de las células, este punto o umbral de cambio se obtiene calculando el valor medio de inducción de la población de células más su respectiva desviación estándar (ver sección de detalles - inicialización).

6.3.2 Diseño de conceptos

Emergencia: la sincronización en la fluorescencia de las células es una propiedad que depende en gran medida de los valores de los parámetros utilizados así como de la densidad celular en la simulación. La inducción es una característica propia de cada célula sin embargo los niveles de

autoinductor extracelular modifican este valor por lo que un aumento de autoinductores provoca también el aumento de la autoinducción individual de las células, que en conjunto se ve reflejado en la fluorescencia sincronizada de la población celular.

Adaptación: la característica adaptativa presente en este modelo se le considera a la regulación de la inducción y producción de autoinductor extracelular, ya que al aumentar la cantidad de autoinductor también aumenta la inducción de las células, sin embargo este aumento de la inducción posteriormente genera una rápida caída del autoinductor y como consecuencia una posterior disminución de la inducción de las células.

Objetivos: los genes tienen las características de que compiten unos contra otros debido al juego de piedra papel y tijeras, sin embargo el objetivo es que la población se mantenga estable sin que desaparezca una de las estrategias empleadas (colores) a lo largo de la simulación. Para ello se aplica la teoría de juegos y se establece una densidad poblacional alta para mantener las oscilaciones variables de cada una de las estrategias empleadas.

Percepción: cada una de las células censa la inducción de las otras células así como los niveles de autoinductor extracelular. Esto afecta su propio valor de inducción para producir la fluorescencia relacionado con lo que se ha visto en la emergencia sobre la inducción sincronizada en la población celular.

Interacción: hay muchos tipos de interacción dentro del modelo. La interacción directa se da en el juego de piedra, papel y tijeras entre las estrategias génicas ya que cada estrategia detecta el color de la otra estrategia que se encuentre en la vecindad. Una interacción indirecta se da entre las células ya que estas censan la autoinducción de sus células vecinas y usan esa información para establecer su propia inducción.

Aleatoriedad: el modelo tiene cierto grado de aleatoriedad ya que al tener sólo 6 juegos de genes, la población de células tiene cierta probabilidad de usar cualquiera de estos juegos de genes durante la simulación, ya que se trata de buscar la mayor variabilidad genética entre células para comprobar que la comunicación por medio de la percepción por quorum puede generar un mecanismo de sincronización en las células.

Observación: son necesarias las gráficas de la dinámica de las estrategias génicas para determinar que estas oscilan y que se mantienen estables independientemente del efecto de la inducción. Igualmente el conteo promedio de inducción dentro de la población y la cantidad de células que adquieren fluorescencia es importante para establecer el parámetro que determine el umbral de cambio para presentarse la fluorescencia. Por último el histograma de frecuencias de las células que son inducidas a fosforescer es importante para determinar si hay sincronización en la población de células.

Inicialización

Los parámetros con los cuales se inicia el modelo son: número de agentes de cada tipo: células (3 para el modelo individual y un máximo de 3000 para el modelo poblacional), genes (600); valores de las constantes: k_{diff} (0.3), k (20), k_{se} (0.3), k_{s0} (0.3), k_{s1} (0.3), n (0.3) (ver tabla 4). El parámetro control de la fluorescencia es de 48 + para cambiar de gris a amarillo y de 45 - para volver de amarillo a gris.

Datos de entrada

Este modelo no utiliza datos de entrada que representen procesos variables en el tiempo.

Submodelos

Se consideran 2 submodelos mayores los cuales se subdividen en varios submodelos inferiores (Figura 16).

El primer submodelo corresponde a la elaboración del proceso de “setup” o inicialización del modelo, donde se establecen las características tanto de los agentes como del mundo. Este submodelo se puede subdividir en tres secciones importantes: la creación de las células, la creación de los genes y la creación del autoinductor. La primera sección se refiere al establecimiento de las células en la celda mayor del mundo. Se define su forma (shape), su tamaño, color y coordenadas (esta parte es importante para determinar su región en la celda mayor) y la cantidad que serán creadas por medio del comando “Número de células” el cual relaciona el valor del slider de la interfaz con la cantidad a ser creada. La segunda sección es muy similar a la anterior pero más compleja ya que requiere también definir los límites entre las celdas por medio de cambios en los colores de los patches de tal forma que formen líneas horizontales y verticales blancas y aquí es sumamente importante establecer correctamente las coordenadas para que se formen 6 celdas menores donde estén los genes y una celda mayor donde se establezcan las células y el autoinductor. Básicamente aquí se emplea el comando “set” para establecer cada una de las características anteriores y una parte interesante es el establecimiento de los colores (rojo, verde y azul) de forma aleatoria en la población de genes. Para esto se emplea el comando “mod” con el cual se puede dividir el número total de genes definido por “genes” de tal forma que el residuo de esta división sea 3, por lo que la población total se divide entre 3 y con el comando “ítem” a cada fracción de la población se le otorga uno de los 3 colores definidos por la variable global colors. En la tercera sección simplemente se define la forma que tendrán los

autoinductores definidos por la forma “ai” la cual es editada por medio del editor de tortugas del programa.

El segundo submodelo corresponde a todos los procesos dinámicos por medio del botón “go”. Se subdivide a su vez en 6 secciones diferentes: 1) el submodelo del movimiento con rebote o “bounce”; 2) el submodelo de la dinámica de la población de genes o “play”; 3) el submodelo para observar la inducción de cada célula; 4) el submodelo de difusión del autoinductor fuera y dentro de la célula (establecimiento de la inducción); 5) el submodelo del efecto de la comunicación por percepción por quorum en el sistema del represilador (población de genes) y 6) el submodelo de la fluorescencia.

En la primera sección se establece un código para generar movimiento aleatorio de los agentes usando primitivas como “heading” y “fw” para cambiar y establecer la longitud del movimiento y usando comando como “set heading” y “if patch ahead” para dar la condición de cuándo deban cambiar su dirección (al chocar con una barrera blanca).

La segunda sección corresponde a la aplicación de la teoría de juegos a la dinámica del represilador establecido por Elowitz (Binmore, 2007). En un principio se pensó emplear tanto la teoría de juegos como la ecuación replicadora siguiendo una dinámica evolutiva (Nowak, 2006), sin embargo el emplear una matriz de pagos y aplicar como tal la ecuación replicadora donde contempla la adecuación y las frecuencias poblacionales de las estrategias no logró la dinámica deseada de oscilaciones por parte de la población de genes en las simulaciones. Por esto se determinó este procedimiento describiendo simplemente las reglas del juego de piedra papel y tijeras y se estableció una condición inicial con una población alta de genes para que las 3 estrategias diferentes se mantuvieran estables. Se emplearon principalmente las primitivas “if” y “let” para definir una variable temporal de reconocimiento del agente vecino de las estrategias génicas siguiendo las ideas de autómatas celulares (Bennett, 2010) y a partir del reconocimiento de este agente en la vecindad se determina el nuevo estado del agente o gen (su coloración) siguiendo las condiciones establecidas por la primitiva “if” y esto se realiza cada vez que es interpretado este procedimiento.

La tercera sección se refiere simplemente al procedimiento para observar o cuantificar el nivel de inducción de las células. Se usa un botón para activar o desactivar esta opción y se emplea la primitiva “label” para etiquetar a cada una de las células con su nivel de inducción respectivo.

La cuarta sección, es quizá la parte más compleja de todo el modelo, ya que en esta parte se incorporan las ecuaciones del modelo de sincronización del represilador mediante la percepción por quorum (García-Ojalvo *et al.*, 2004). Dado que tenemos 6 diferentes juegos de genes se requieren 6 diferentes procedimientos para establecer la dinámica del autoinductor y la inducción de manera que se emplee la información de cada uno de estos juegos de genes. En general los procedimientos empleados son iguales para la dinámica del autoinductor y la inducción respectivamente, lo único que cambia es la información que se extrae de cada una de las celdas que subdividen a la población de estrategias génicas. Más específicamente las coordenadas que delimitan cada celda son básicamente el factor que diferencia a los 6 procedimientos. Las ecuaciones matemáticas se describen de forma semántica, sin embargo Netlogo tiene un amplio código matemático que permite realizar operaciones matemáticas con lo cual, se pueden describir diversas funciones. Los valores de las constantes empleadas son definidos en la interfaz del modelo, los términos de concentraciones y las variables que cambian en el tiempo en las ecuaciones (como S_i y S_e) fueron definidas empleando la primitiva “let” para definir el valor de estos temporalmente y el comando “set” para establecer el valor de inducción y de autoinductores respectivamente una vez evaluado el cambio de cada una de las concentraciones, en este caso del conteo de agentes (autoinductores o genes) después de cada paso discreto o tick. La manera en que se recopila la información de la inducción es preguntándole a las tortugas el valor de inducción de sus células vecinas, tal como lo haría un autómata celular (Deutsch y Dormann, 2005). Este valor se emplea como la concentración de autoinductor interno de la célula (inducción) y se evalúa con los procedimientos relacionados con las ecuaciones. Este nuevo valor adquirido se le asigna aleatoriamente a las células con lo cual la variabilidad de los juegos de genes se ve reflejada en su inducción y posterior fluorescencia a lo largo de la simulación.

La quinta sección corresponde al efecto de las moléculas de quorum sensing extracelulares en la actividad génica, que podría decirse es el proceso inverso de la sección anterior. En este caso, se recopila la información de la inducción de las células y se evalúa el cambio en la población de genes de los 6 juegos de genes con respecto a la información obtenida de la inducción celular. Aquí también se usan los comandos de establecimiento “let” y “set” para establecer los nuevos valores de la población de las células con color azul, que corresponden a las estrategias génicas c_i .

La última sección corresponde al establecimiento de un criterio o parámetro control para definir cuando las células pasan de un estado de no fluorescencia (coloración gris) a uno con presencia de fluorescencia (coloración amarilla). Dado que en la literatura no existe un consenso de una cantidad o cuantificación de la expresión de las moléculas responsables de generar la fluorescencia se hizo el supuesto de que el autoinductor es la molécula responsable de generar la cascada de señalización para generar la fluorescencia, por lo que el nivel de inducción de cada célula sería el parámetro para definir la fluorescencia. Así altos niveles de inducción generarían fluorescencia en la célula y bajos niveles regresarían a las células a su estado normal. Para esto se

calculó la media del nivel de inducción de las células al igual que su desviación estándar. Con esto se obtuvo un umbral de cambio para presentar fluorescencia y otro punto umbral para regresar al estado normal de las células.

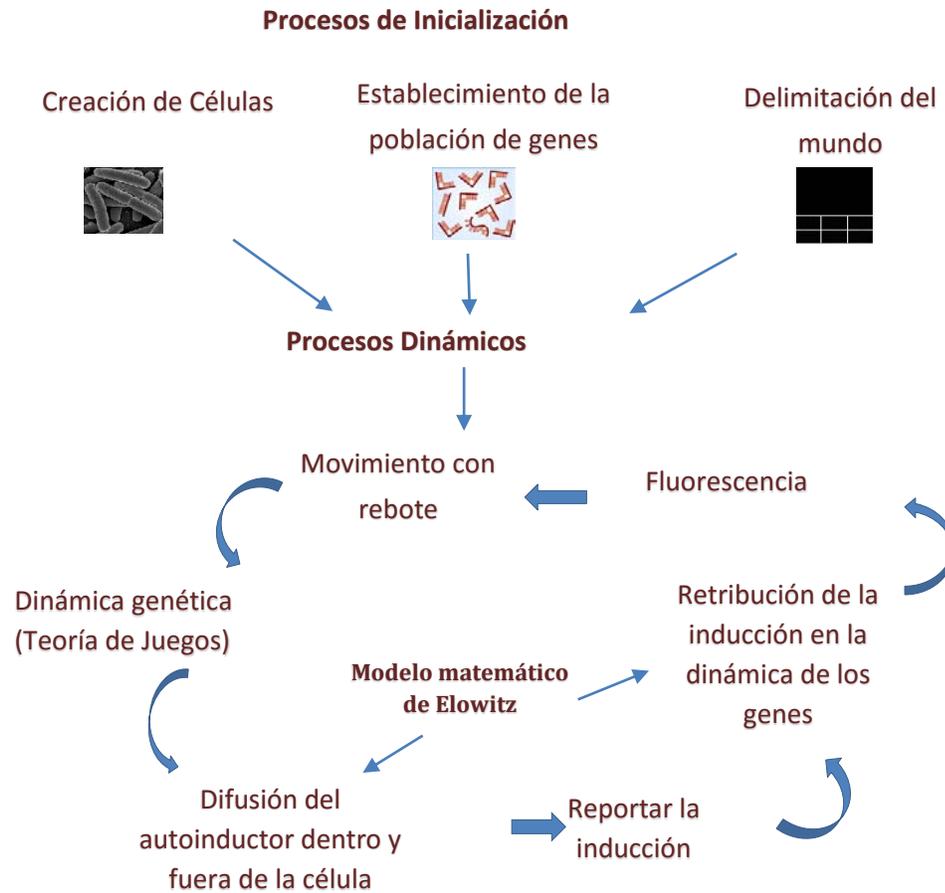


Figura 16. Diagrama simplificado de la serie de procesos que se llevan a cabo en el modelo de Netlogo.

6.4 Llevar a cabo las simulaciones

Se corrió el modelo en Netlogo versión 5.1.0 en una computadora HP 240 G3 con sistema operativo Windows 8.1. Se llevaron a cabo 100 corridas y los datos de las gráficas e histograma del modelo se guardaron en tablas de Excel.

Parte III

Resultados

7. Resultados

Se obtuvieron 2 modelos computacionales en Netlogo que reflejan el comportamiento de sincronización de bacterias mediado por el sistema de quorum sensing acoplado al sistema del represilador de Elowitz. El primer modelo contempla una población celular estable de 3 células con sus respectivos juegos de genes correspondientes (Fig. 17). El otro modelo considera una población celular mayor, donde cada célula posee una actividad genética particular, dados 6 diferentes juegos de genes que aleatoriamente son empleados por cada una de las células para determinar su inducción y producción de autoinductor extracelular. Este último modelo introduce el concepto de densidad celular como un parámetro que controle la sincronización por comunicación extracelular (Fig. 18). Los parámetros y condiciones iniciales mostraron ser determinantes para la generación de oscilaciones en la población de genes y en la fluorescencia. Los parámetros fueron probados llevando a cabo la simulación del modelo matemático de Elowitz en un programa de resolución de sistemas de ecuaciones (Polymath) para evaluar el modelo matemático.

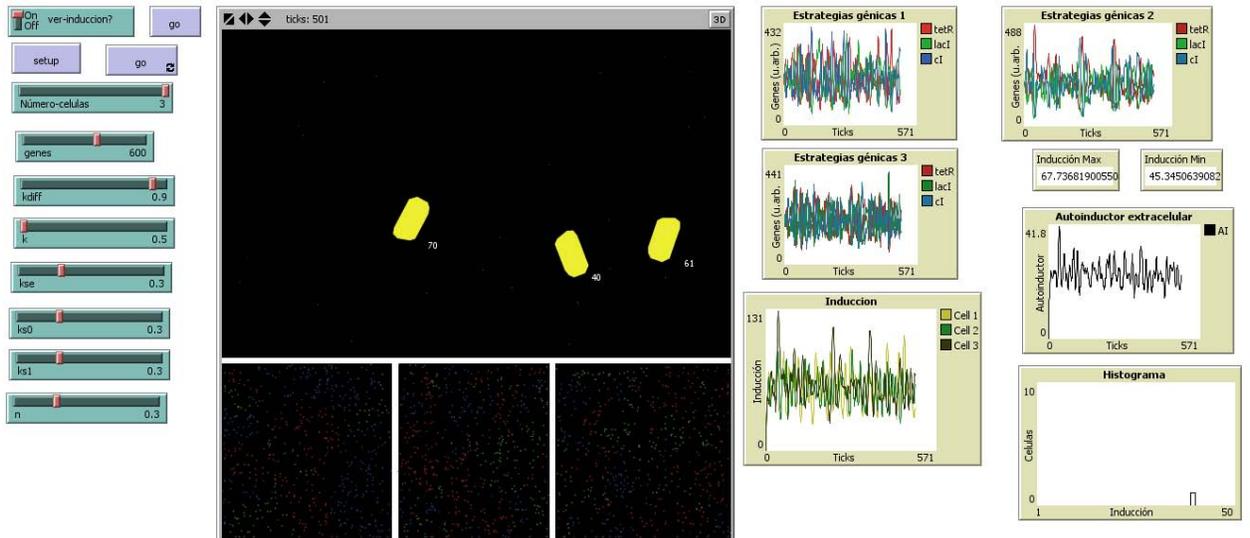


Fig. 17 Primer modelo de sincronización sin densidad celular. En este primer modelo solo hay 3 células presentes y 3 juegos de genes para cada una de las células. Las gráficas de los genes y de la fluorescencia muestran oscilaciones del tipo sinusoidales.

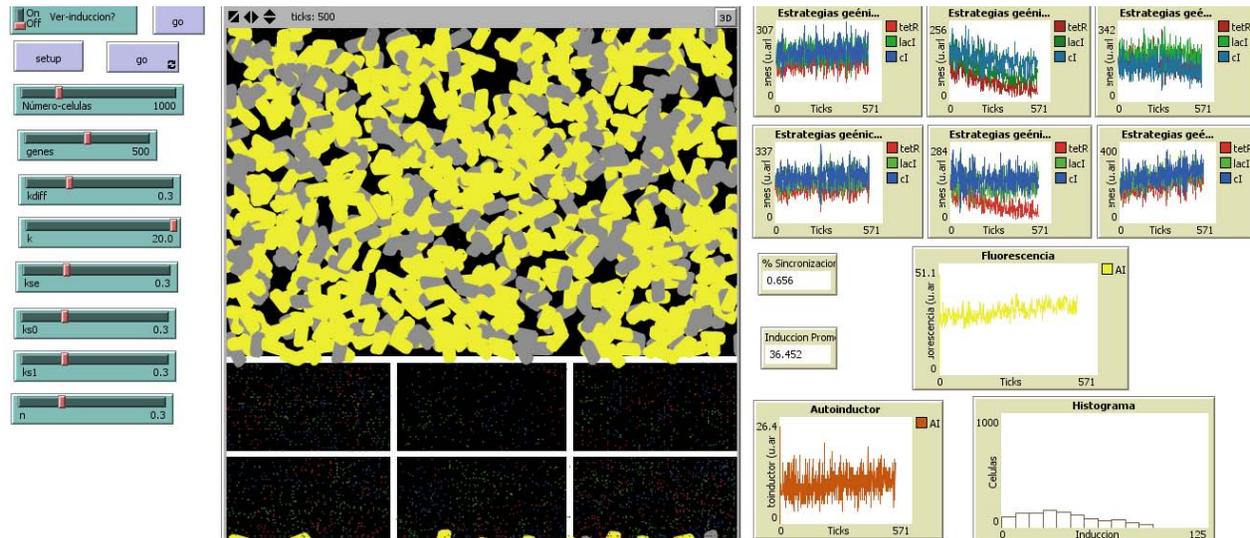


Fig.18. Segundo modelo donde se incorpora la densidad celular para evaluar si ésta actúa como criterio para la generación de sincronización en los niveles de fluorescencia.

7.1 Modelos en Netlogo

Los modelos obtenidos en Netlogo se plantearon aplicando la teoría de juegos a la dinámica de los genes que forman parte del represilador al seguir una actividad de represión cíclica. Posteriormente se utilizaron las ecuaciones diferenciales de la difusión del autoinductor dentro y fuera de la célula (propuestas por Elowitz) para determinar la actividad del autoinductor en el modelo. Los parámetros y condiciones iniciales que se establecieron fueron los mismos para ambos modelos. Se observaron oscilaciones en la fluorescencia así como en la actividad génica de las tres estrategias génicas. Las oscilaciones en las estrategias génicas se muestran similares cualitativamente en el modelo de 3 líneas celulares y en el modelo poblacional (Fig. 17 y Fig.18) sin embargo se aprecia una mejor sincronización de la fluorescencia en el modelo poblacional. El comportamiento de las gráficas de la concentración del autodincutor extracelular en el modelo de 3 líneas celulares también se muestra semejante al modelo poblacional.

7.1.1 Modelo de sincronización en tres líneas celulares

En este tipo de modelo solamente se sigue la línea temporal de 3 células con sus respectivos juegos de estrategias génicas. Las simulaciones computacionales indican que la población de estrategias génicas alcanza un equilibrio estable siempre y cuando la población sea bastante grande. Se encontró que cuando tiene una población menor a 500 células se pierde el equilibrio en las oscilaciones de la población de cada una de las estrategias y éste se desplaza hacia una de las estrategias empleadas, por lo que las demás estrategias son eliminadas de la población (Fig. 19a). Se aplicó la teoría de juegos directamente en las reglas de interacción entre los agentes para determinar el comportamiento de los genes y la frecuencia de cada una las estrategias fue de 0.3 ya que la población total de estrategias génicas establecida por el slider se dividía entre 3, logrando así la estabilidad de la población a lo largo de la simulación (Figura 19b). El promedio de la población de estrategia génica roja para 100 simulaciones y con una población de 500 estrategias génicas fue de 160 con una SD de $3.770 \pm$, para la estrategia verde fue de 220 con una SD de $4.120 \pm$ y para la estrategia azul fue de 120 con una SD de $3.590 \pm$.

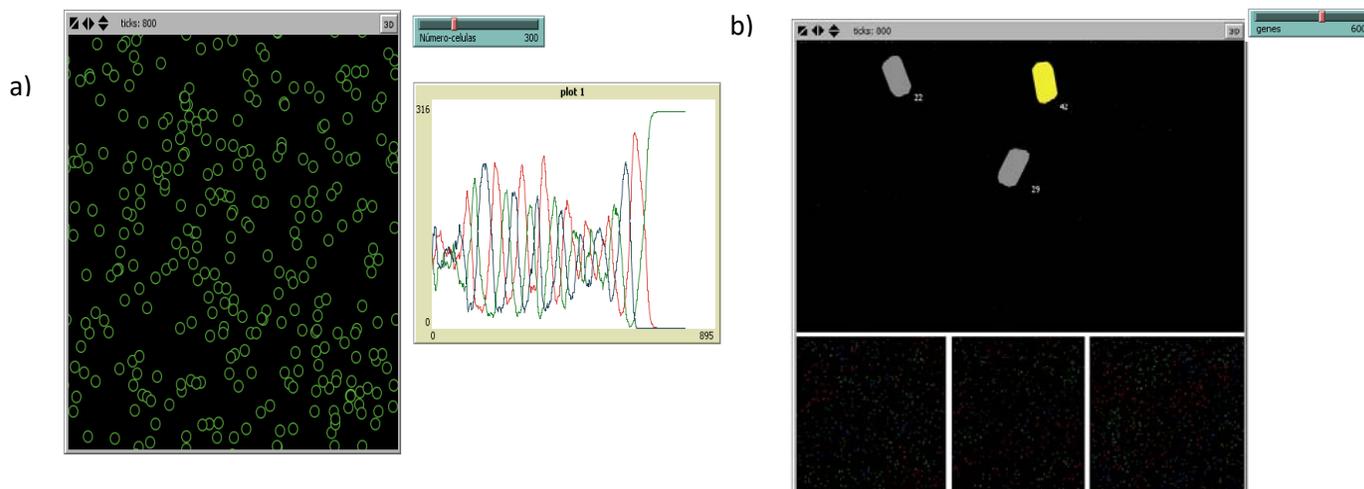


Figura 19. Imagen de la simulación de la población de genes donde la densidad poblacional juega un papel importante para mantener una dinámica estable de la frecuencia de cada una de las estrategias. a) Población de 300 mRNA después de 800 ticks. B) Población de 600 mRNA en cada parcela después de 800 ticks.

Las ecuaciones del modelo de Elowitz (Garcia-Ojalvo et al., 2004) fueron implementadas en el código de Netlogo para regular la producción de autoinductor fuera de la célula así como el autoinductor interno lo que desencadena la señalización para la activación de genes responsables de producir la fluorescencia de la célula. El formalismo de la ecuación se mantuvo prácticamente intacto dado que el programa contiene varias primitivas de carácter matemático, con lo cual fue posible plantear las ecuaciones diferenciales con sus respectivas operaciones aritméticas (ver anexo del código). La parte importante a implementar dentro del programa, fue el establecimiento del código para calcular el cambio con respecto al tiempo de las concentraciones de AI, ya fuera en las células individuales o considerando la población total. La ecuación del autoinductor intracelular corresponde a una ecuación que determina el cambio en la concentración del autoinductor interno para cada célula individual (inducción). Mientras que la ecuación de la dinámica del autoinductor extracelular determina el cambio en la concentración del autoinductor fuera de la población de células, tomando en cuenta la concentración de autoinductor intracelular (inducción) poblacional (ver sección de Métodos). Por ello la aplicación de una ecuación individual, es decir, para cada uno de los agentes y otra ecuación para establecer una población de autoinductores, considerando la información de la variable propia de las células (inducción), generó el comportamiento oscilatorio de la fluorescencia. Las gráficas de la fluorescencia muestran que la amplitud de las oscilaciones tiene un umbral de oscilación y no aumenta o decrece con el paso del tiempo. Sin embargo las gráficas parecen estar desorganizadas y no hay un ritmo colectivo bajo estas condiciones de baja población celular (Fig. 20). El promedio

de los valores máximos de inducción para la célula 1 fue de 31.218 con una SD de ± 6.282 , para la célula 2 fue de 28.752 con una DS de ± 6.105 y para la célula 3 fue de 35.600 con una SD de ± 6.609 .

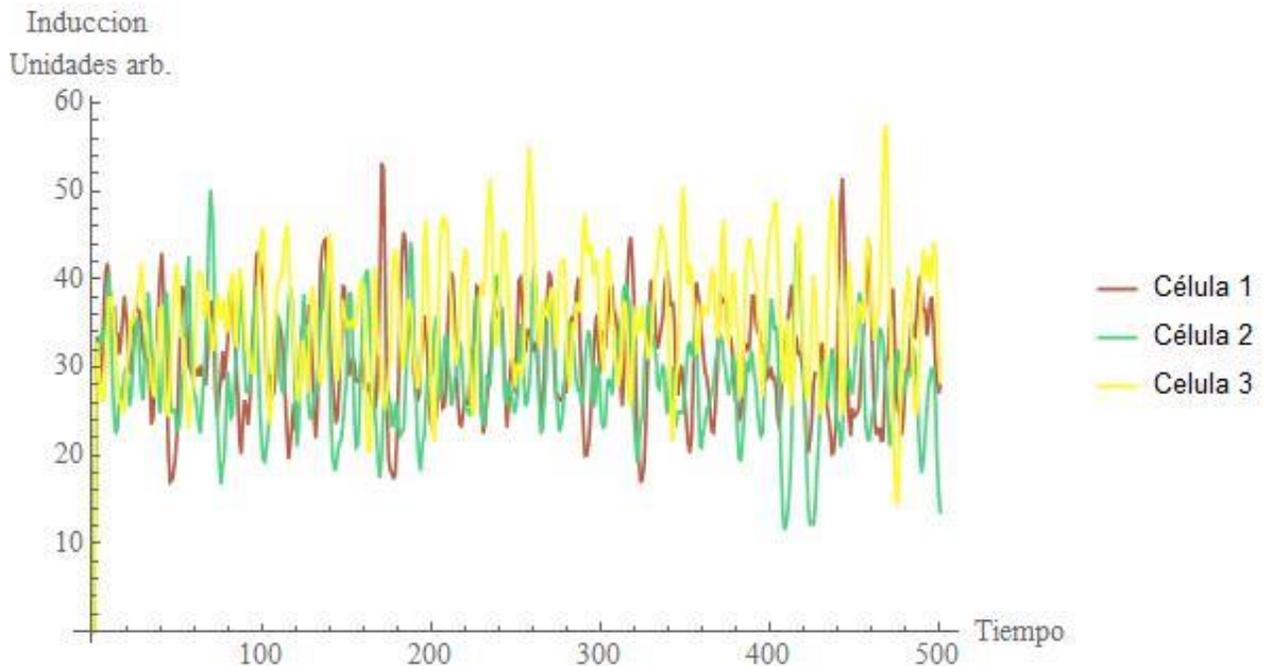


Figura 20. Grafica de la inducción o autoinductor (AI) intracelular de las 3 líneas celulares en una simulación de 500 ticks.

Los parámetros de las constantes cinéticas de las ecuaciones del modelo de sincronización de Elowitz fueron factores que afectaron enormemente el comportamiento de la inducción y producción de autoinductor extracelular. En general cualquier valor cercano a la unidad ocasionaba un aumento drástico de estos valores de autoinducción y de autoinductores extracelulares. Esto generaba un exceso de procesamiento de información en el programa reduciendo su rendimiento o en ocasiones deteniendo su funcionamiento en su totalidad. Por ello fue necesario realizar simulaciones de las ecuaciones del modelo de Elowitz para determinar cómo afectaban estos parámetros al comportamiento de los genes y del autoinductor interno y externo de la célula (ver Discusión). Los parámetros que se emplearon en las simulaciones se observan en la Tabla 3.

Parámetro	Valor	Significado biológico
<i>Células</i>	3	Este valor corresponde a la cantidad de células bacterianas que se pretende simular en el modelo computacional.
<i>Estrategias génicas</i>	500	Este valor corresponde a la cantidad de genes o la población bacteriana que se pretende simular en el modelo donde el juego se da en los genes o en el modelo donde se da a nivel de las estrategias celulares respectivamente.
<i>Kse</i>	0.3	Es una constante cuyo valor corresponde a la tasa de degradación del autoinductor extracelular.
<i>Ks0</i>	0.3	Es una constante cuyo valor corresponde a la tasa de degradación del autoinductor intracelular.
<i>Ks1</i>	0.3	Es una constante cuyo valor corresponde a la tasa de síntesis del autoinductor intracelular.
<i>K</i>	20	Es una constante cuyo valor corresponde a la contribución que genera la presencia de saturación del AI a la transcripción de <i>lacI</i>
<i>Kdiff</i>	0.3	Es una constante que cuyo valor corresponde a la tasa de difusión del autoinductor fuera de la célula, en relación con el volumen extracelular.
<i>n</i>	0.3	Es una constante cuyo valor corresponde a la tasa de difusión del autoinductor a través de la membrana celular desde o hacia el medio intercelular.

Tabla 3. Valores usados de los diferentes parámetros de los modelos de Netlogo.

El valor de n fue quien más afectó el comportamiento de las oscilaciones y principalmente la regulación del autoinductor intra y extracelular, ya que si los valores de esta constante eran muy altos, las concentraciones de AI tanto interno como externo se incrementaban enormemente durante la simulación. Se empleó la herramienta de Netlogo analizador de comportamiento para observar qué valores de los parámetros afectaban en gran medida el comportamiento de las oscilaciones, ya fuera de la población de genes, de la inducción o de la fluorescencia (ver Discusión y Conclusiones). Estos valores no correspondieron del todo a los valores utilizados en el trabajo de Elowitz y el comportamiento de las oscilaciones en los genes es más parecido al modelo estocástico del represilador que al modelo determinístico continuo (Elowitz, M & Leibler, S. 2000).

Las oscilaciones en la población de estrategias génicas son variables en los tres diferentes juegos. Cada una de las estrategias génicas empleadas coexiste con las otras estrategias sin que una llega a desaparecer. La población de estrategia génica que emplea el gen *lacI* es quien presenta valores más altos en la amplitud de las oscilaciones ya que la población de estrategia génica *lacI* es la parte que es modulada por el acoplamiento por quorum. Los valores máximos en la amplitud de las oscilaciones en la población de estrategias génicas para el juego 1, para el juego 2 y para el juego 3 fueron respectivamente: para la estrategia roja fue de 349, 322 y 403; para la

estrategia verde fue de 393, 449 y 410; y para la estrategia azul fue de 246, 219 y 299. Mientras que los valores mínimos para estos 3 juegos fueron respectivamente: para la estrategia roja 79, 59 y 99; para la estrategia verde fueron 121, 127 y 131; y para la estrategia azul fueron 73, 50, 57. Estos valores indican que la amplitud de las oscilaciones en la estrategia verde fueron las mayores de las 3 estrategias génicas mientras que la amplitud de las oscilaciones de las estrategias génicas azul fueron las menores (Fig. 21). Esto debido a que la estrategia verde corresponde a la dinámica del gen *lacI* el cual se ve afectada por el sistema de quorum sensing al promover la transcripción del gen debido a las moléculas de autoinductor.

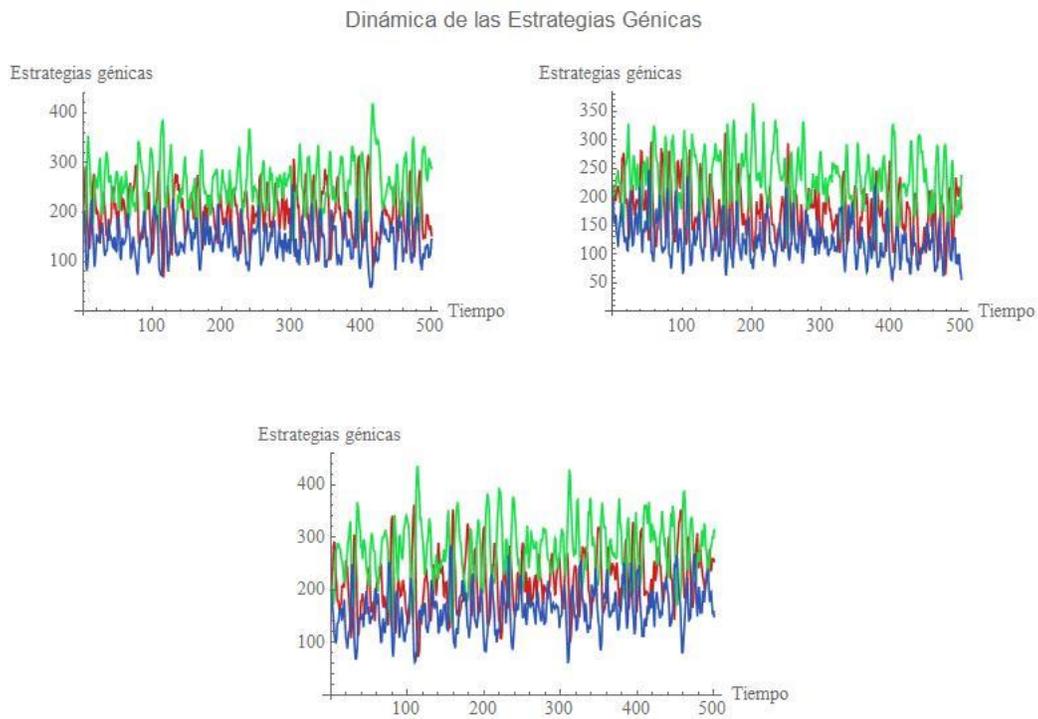


Figura 21. Gráfica de la dinámica poblacional de los 3 juegos de genes del modelo de 3 líneas celulares. Las oscilaciones se mantienen constantes a lo largo de la simulación y su amplitud es variada. Verde corresponde a *lacI*, rojo a *tetR* y azul a *cl*.

7.1.2 Modelo de sincronización poblacional

Dado que la densidad celular cumple un papel importante en la regulación de la sincronización por percepción por quorum (Ward et al., 2001), se modificó el modelo anterior donde la densidad poblacional no era posible modificarla, ya que solo se sigue la línea temporal de 3 células y estas a su vez tienen asignado un juego de genes particular (Fig. 16). Para ello se modificó este modelo para poder aumentar la población y observar los efectos del acoplamiento, sin embargo se encontró el inconveniente de que sería necesario también diseñar más juegos de genes para cada una de las células extras, por lo que la simulación se hacía más complicada y tardada. Para esto se decidió duplicar el juego anterior de genes a 6 y se modificó el código de Netlogo para designar aleatoriamente alguno de estos juegos de genes a cualquiera de las células (ver anexo código), así la simulación no se cargaba de tanta información y podía correr más rápidamente que al crear y asignar un juego de genes para cada célula.

Se emplearon los mismos valores de las constantes y de la población de las estrategias génicas en este modelo, lo único que varió fue la población de células. Se analizó el modelo con 4 diferentes poblaciones celulares: 500, 1000, 2000 y 3000 células. En este nuevo modelo el comportamiento de las oscilaciones tanto de la población de genes como de la fluorescencia no cambió drásticamente comparándolo con el modelo de tres líneas celulares (Fig. 22). Las gráficas de la dinámica poblacional de las estrategias génicas muestran comportamientos muy parecidos para las 4 densidades poblacionales. Estas corresponden de arriba abajo a las celdas de estrategias génicas que se observan en la figura 18 de las celdas superiores de izquierda a derecha. El promedio de valores máximos y mínimos de las oscilaciones de las 6 juegos de estrategias génicas para las 4 densidades poblacionales se observa en la Tabla 4. Estos valores se asemejan a los obtenidos por el modelo de 3 líneas celulares, en donde la población de estrategias verdes (que corresponden a la estrategia *lacI*) mantenía la mayor amplitud de las oscilaciones sin embargo en este caso las oscilaciones de la estrategia *tetR* menor (color rojo) tienen los valores más bajos en comparación con el modelo de 3 células, esto concuerda más con la dinámica de piedra papel y tijeras del represilador ya que si la población de estrategias verdes es mayor, la población de estrategias rojas debería de disminuir pero sin llegar a desaparecer (Fig. 22).

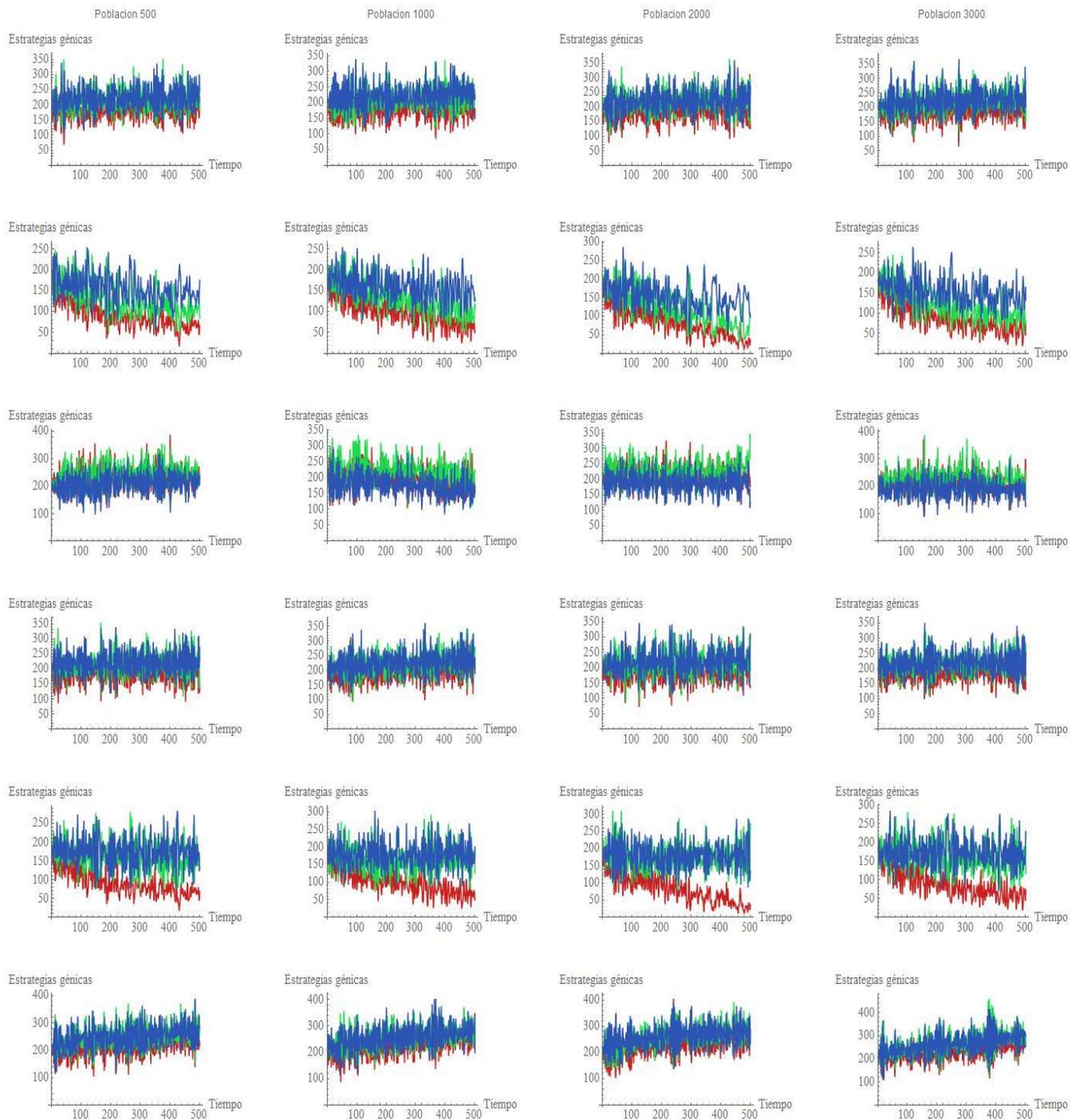
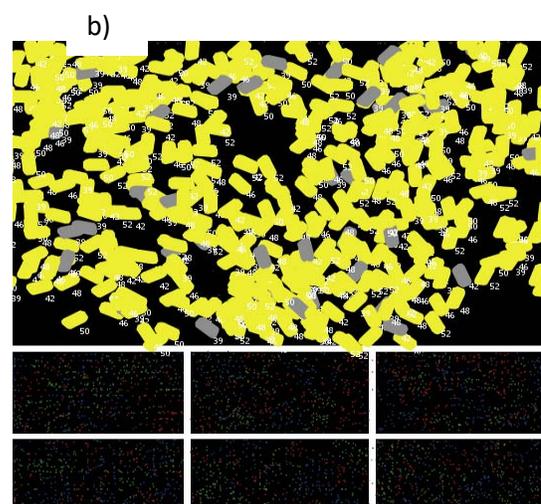
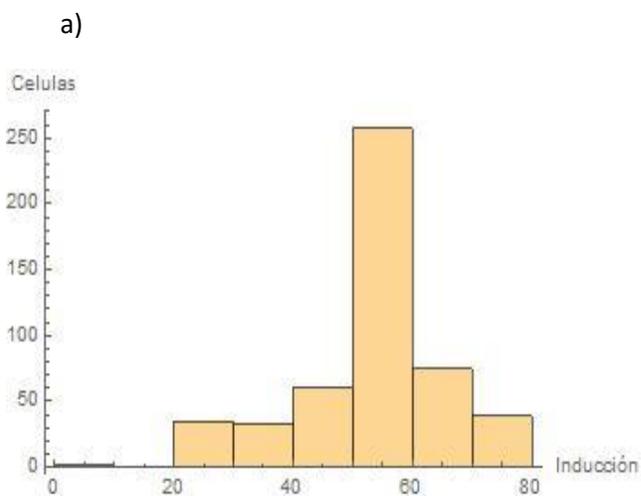


Figura 22. Grafica de la Dinámica Poblacional de los 6 juegos de genes del modelo de sincronización poblacional. Los colores corresponden a los empleados en la gráfica 17.

	Población Celular 500		Población Celular 1000		Población Celular 2000		Población Celular 3000	
	Valor Máximo	Valor Mínimo	Valor Máximo	Valor Mínimo	Valor Máximo	Valor Mínimo	Valor Máximo	Valor Mínimo
Estrategia Roja	294	68	280	70	291	66	307	67
Estrategia Verde	330	101	320	95	332	93	341	92
Estrategia Azul	320	93	326	101	327	192	331	96

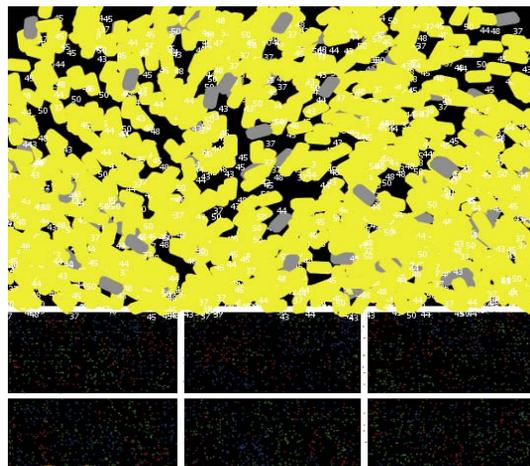
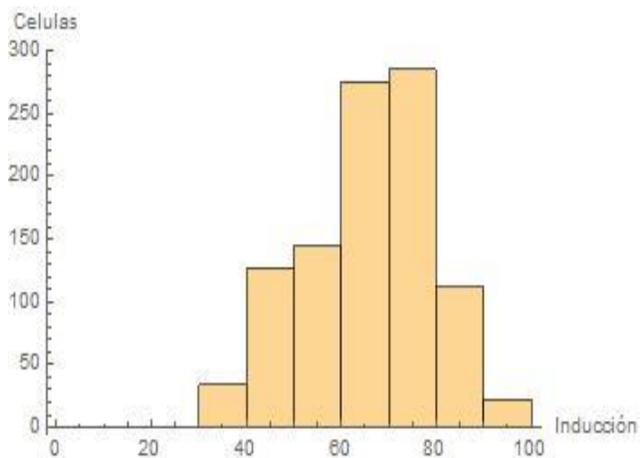
Tabla 4. Valore máximos y mínimos de oscilación de la población de estrategias génicas del modelo de sincronización poblacional.

Los histogramas de inducción muestran que si la población celular aumenta hay mayor número de células en estado de inducción, por lo que se observa la fluorescencia de manera uniforme (Fig. 23). El umbral de inducción se estableció calculando la media poblacional de 100 simulaciones con una población de 500 células obteniéndose un valor de 48 con una SD de 8.048. El incremento en la densidad poblacional generó un aumento en el porcentaje de células que alcanzaban este umbral de inducción, produciendo fluorescencia. Con una población de 500 a 1000 células del 74 - 82.3% de las células se encuentra en un estado de inducción que genere fluorescencia. En una población de 2000 a 3000 células el 82.3 - 85.9 % de las células llegan al umbral de inducción para generar fluorescencia. Los parámetros utilizados fueron los mismos que los empleados en el modelo anterior, lo único que se modificó fue la población celular.

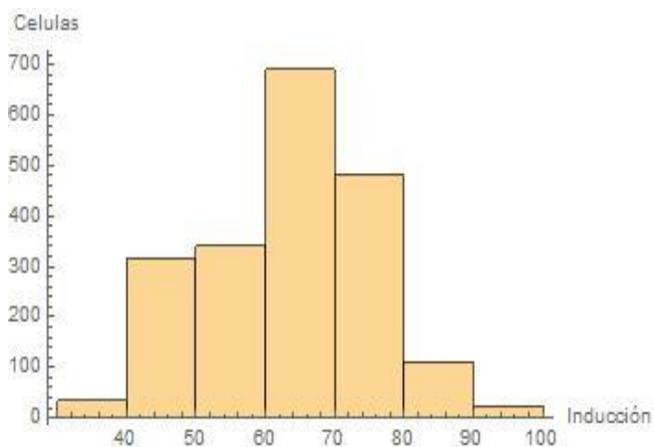


c)

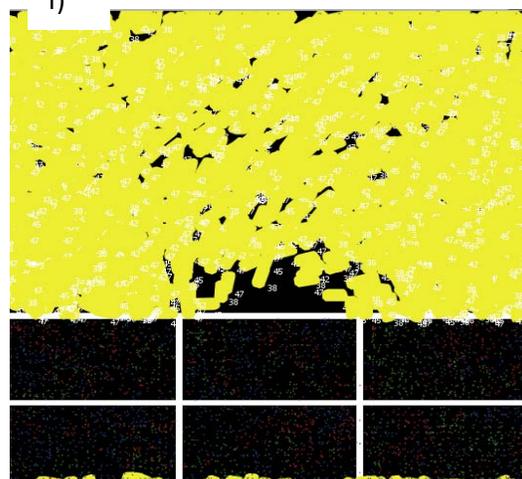
d)



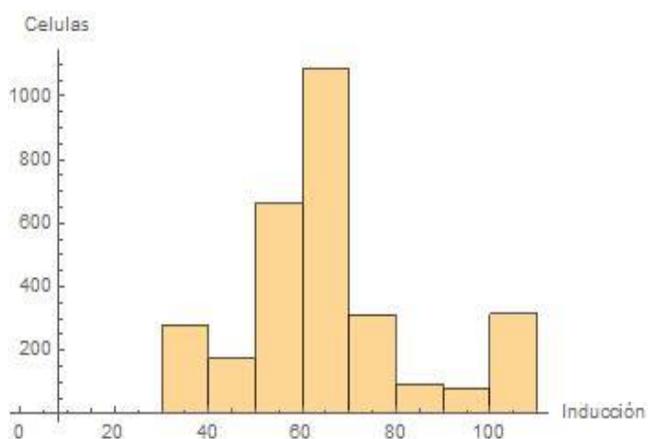
e)



f)



g)



h)

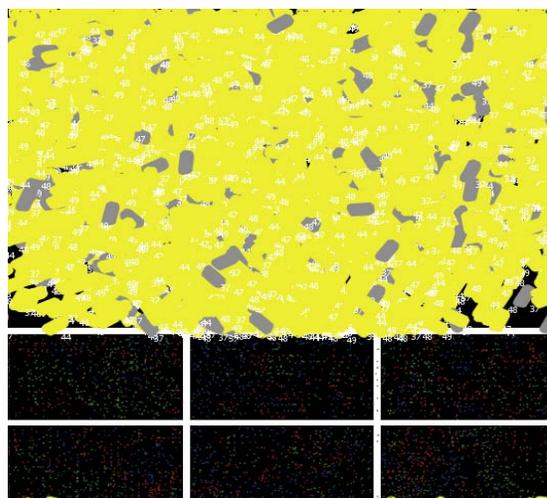


Figura 23. Histograma de los niveles de inducción en las células (a,c,e y g) e imágenes de la sincronización de las simulaciones en Netlogo (b,d,f y h) a diferentes niveles poblacionales. En a) y b) la población de células es de 500. En c) y d) la población es de 1000. En e) y f) la población es de 2000. En g) y h) la población es de 3000. Los parámetros de las constantes corresponden a los empleados en la tabla 3.

Se observó que también aumenta el grado de inducción conforme aumenta la población celular así como el autoinductor producido por las células. En la figura 24 se observan las gráficas de los promedios de inducción de la población total de células en una corrida de 500 ticks y a varias densidades poblacionales. Este valor varió en las diferentes simulaciones llevadas a cabo pero en general se observa un aumento en este valor promedio de inducción conforme se aumenta la población y concuerda con los histogramas de inducción donde la mayoría de la población celular alcanza el nivel de inducción necesario para producir la fluorescencia. También se observa que el aumento de la población genera mayor amplitud en las oscilaciones de la inducción mientras que en una baja población celular esta amplitud no es tan alta.

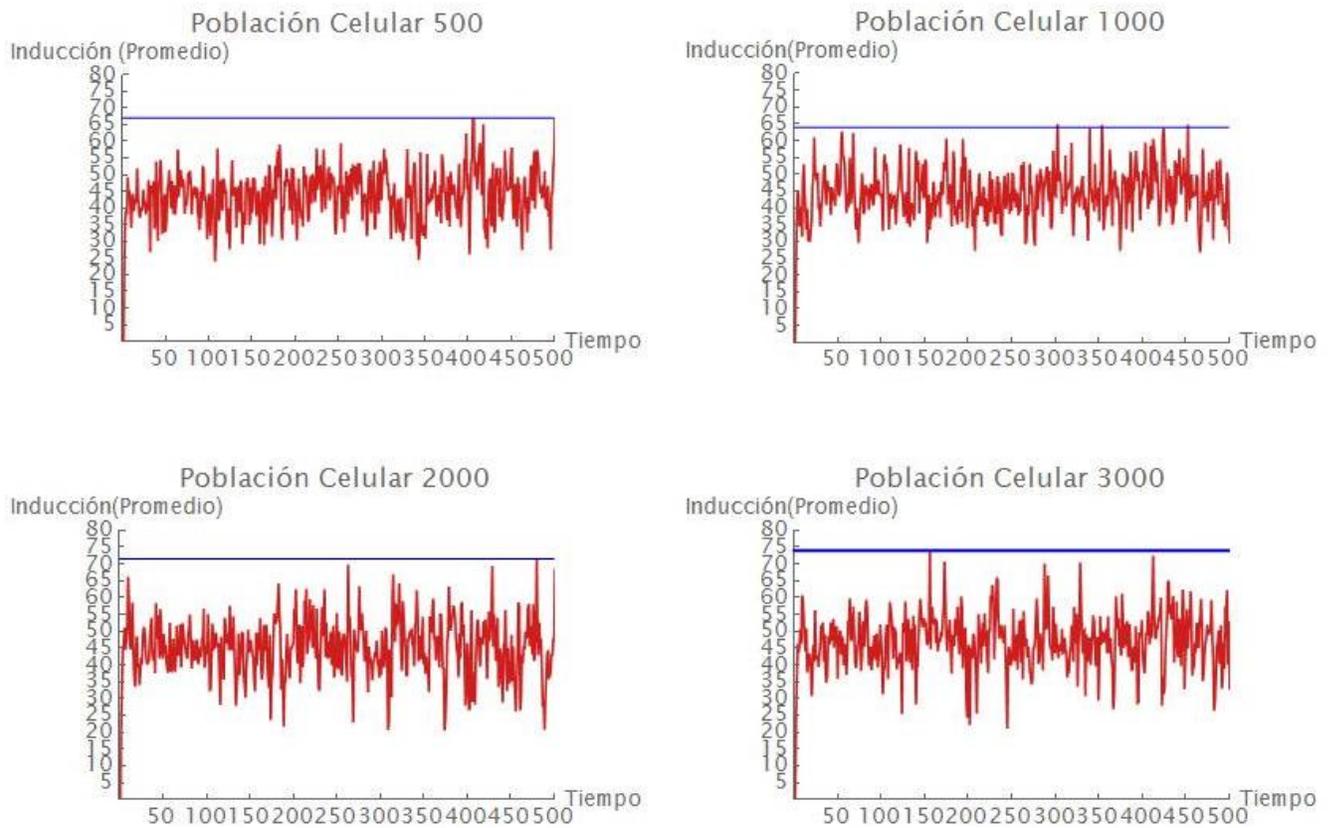


Figura 24. Graficas de la inducción promedio en una simulación de 500 ticks a diferentes densidades poblacionales. Las líneas azules horizontales corresponden a los valores máximos de la inducción promedio y este valor aumenta ligeramente conforme se aumenta la población celular.

Se realizó la verificación de las simulaciones siguiendo el enfoque que dan Law y Kelton (Law & Kelton 2003). Se siguieron 5 técnicas para el desarrollo del código y simulación: 1) se corroboraron los submodelos en módulos o como subprogramas desarrollando primero un modelo simple y sencillo con el cual se pudiera trabajar obteniendo el comportamiento deseado y permitir representar más fielmente el sistema a modelar (ver anexo 2). En el desarrollo del programa se discutió la simulación con otras personas involucradas en la programación y modelación de fenómenos biológicos para plantear posibles problemas o modificaciones necesarias en algunos de los procedimientos ejecutados durante la simulación y aclararlos o corregirlos antes de continuar con los siguientes procedimientos del modelo. 3) El empleo de indicadores (el valor de un estado de variable, algún valor estadístico o la interpretación del mismo código) imprimiendo este resultado en la terminal o en algún gráfico de salida inmediatamente después de cada paso temporal o en este caso de los ticks; así se imprimieron los valores que tomarían las nuevas concentraciones de autoinductor interno y externo al igual que los valores de la re-calibración de las poblaciones de estrategias génicas (procedimientos difusión-out, difusión-in y lac-recali), esto para asegurarse que el procesamiento del código se llevó a cabo correctamente. 4) Se corrió la simulación bajo las suposiciones simplificadas de que el sistema de bacterias sería sincronizado por medio de las moléculas de quórum sensing según las ecuaciones de Elowitz con los parámetros de las constantes establecidas en la Tabla 3 y midiendo variables como la población de agentes (genes y autoinductor), la autoinducción y las células con fluorescencia, las cuales son conocidas o son fácilmente computables. 5) Finalmente durante las simulaciones los datos de salida durante cada paso discreto (ticks) se mostró por medio de gráficas, monitores e histogramas para contabilizar las variables medidas según las suposiciones establecidas anteriormente.

Parte IV

Discusión y Conclusiones

8. Discusión

Se revisó el modelo matemático de sincronización por percepción por quorum que Elowitz y col. publicaron en el año 2004 y que consiste en una serie de 7 ecuaciones diferenciales acopladas que representan la difusión del AI, así como la dinámica genética y de las proteínas que regulan el represilador. Las ecuaciones fueron implementadas en el programa Polymath manteniendo los mismos valores de las constantes cinéticas de la tabla 4 excepto por el valor de “k” ya que este valor alteraba significativamente el comportamiento de las oscilaciones. Al ser un modelo de ecuaciones diferenciales se dieron condiciones iniciales de las diferentes sustancias: para el mRNA y las proteínas se dieron una concentración inicial de 20, mientras que para el autoinductor intra y extracelular se dio una concentración de 25. El comportamiento oscilatorio tanto para el mRNA como para el autoinductor intra y extracelular concuerda con el comportamiento observado en el modelo en Netlogo (Fig. 25). La variación en la amplitud de las oscilaciones del mRNA comparado con las oscilaciones de las estrategias génicas son muchos menores pero esto debido a la diferencia en la escala de las concentraciones de ambos componentes. Sin embargo los niveles de S_i y S_e se mantienen por lo bajo de las oscilaciones del mRNA y las estrategias génicas respectivamente y S_e a su vez se mantienen por niveles más bajos que S_i , tanto en las gráficas del modelo matemático como en las gráficas del modelo en Netlogo. Esto sugiere que el comportamiento de la dinámica genética establecida en el modelo basado en agentes concuerda con el comportamiento que predice el modelo matemático. Esto resalta a su vez la importancia de la teoría de juegos en su implementación en modelos basados en agentes para explicar fenómenos biológicos en este caso, la dinámica de un represilador siguiendo las reglas de piedra papel y tijeras.

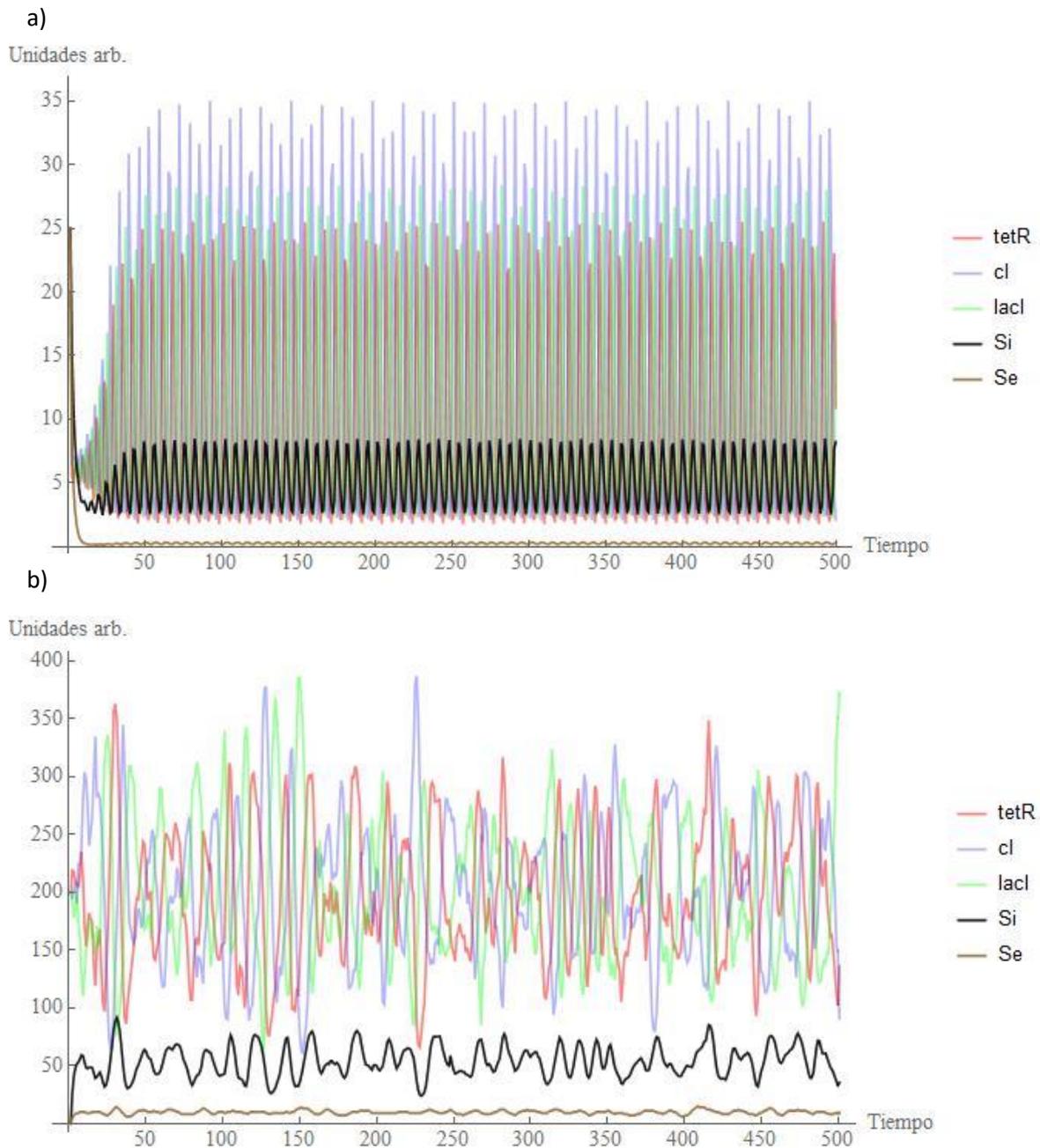


Figura 25. Graficas de los niveles de mRNA y autoinductor intra (Si) y extracelular (Se) así como de la población de estrategias génicas y los niveles de inducción y autoinductores comparando el modelo de Netlogo y las simulaciones en Polymath. a) Grafica de la resolución de ecuaciones para 1 sola célula con una $K = 1$. B) Grafica del modelo de Netlogo del seguimiento de 1 célula con una $k = 1$.

En ambos modelos los valores de las constantes cinéticas y de las tasas de regulación genética jugaron un papel importante para obtener el comportamiento oscilatorio deseado tanto en las estrategias génicas como en los valores de la inducción y la fluorescencia. En las siguientes tablas se muestra el comportamiento de las oscilaciones con diferentes valores de los 6 parámetros empleados en las simulaciones. Cada parámetro se varió en 3 órdenes de magnitud diferentes tomando el mismo valor pero aumentando su magnitud, empezando de menor a mayor valor.

Parámetro K	Valor: 0.2				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	40 - 436	Si	Si	45.5011976
	1000	60 - 411	Si	Si	48.6476327
	1500	55 - 411	Si	Si	51.3110858
2000	49 - 399	Si	Si	55.0150968	

Parámetro K	Valor: 2				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	54 - 376	Si	Si	45.0848623
	1000	42 - 377	Si	Si	47.9500519
	1500	41 - 445	Si	Si	50.7909022
2000	47 - 383	Si	Si	54.1785649	

Parámetro K	Valor: 20				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	16 - 415	Si	Si	42.1381677
	1000	35 - 411	Si	Si	44.7479581
	1500	20 - 392	Si	Si	47.6811484
2000	19 - 418	Si	Si	50.5644321	

Tabla 5. Comportamiento oscilatorio de las estrategias génicas y de los niveles de inducción interno y externo de la célula, así como de los niveles de fluorescencia poblacional al variar el parámetro k.

Parámetro n	Valor: 0.01				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	26 – 425	Si	Si	42.8801238
	1000	23 - 416	Si	Si	43.2347625
	1500	3 - 391	Si	Si	43.7521397
2000	6 - 427	Si	Si	44.2311796	

Parámetro n	Valor: 0.1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	17 - 426	Si	Si	41.7333653
	1000	28 - 419	Si	Si	42.3764072
	1500	8 - 401	Si	Si	43.148833
2000	25 - 416	Si	Si	43.7352585	

Parámetro n	Valor: 1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	14 - 417	No	No	-8.34E+18
	1000	7 - 413	No	No	-8.46E+18
	1500	13 - 414	No	No	-8.54E+18
2000	18 - 368	No	No	-8.62E+18	

Tabla 6. Comportamiento oscilatorio de las estrategias génicas y de los niveles de inducción interno y externo de la célula, así como de los niveles de fluorescencia poblacional al variar el parámetro n.

Parámetro kse	Valor: 0.01				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	35 - 413	Si	Si	42.5366906
	1000	30 - 404	Si	Si	45.1827725
	1500	21 - 391	Si	Si	47.855823
2000	44 - 407	Si	Si	51.3481078	

Parámetro kse	Valor: 0.1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	11- 442	Si	Si	42.3691497
	1000	29 - 419	Si	Si	44.7961996
	1500	20 - 401	Si	Si	47.9558935
2000	34 - 416	Si	Si	51.0528074	

Parámetro kse	Valor: 1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	22 - 413	No	Si	39.8826786
	1000	17 - 402	No	Si	42.7050918
	1500	19 - 404	No	Si	45.5879428
2000	25 - 389	No	Si	48.2207794	

Tabla 7. Comportamiento oscilatorio de las estrategias génicas y de los niveles de inducción interno y externo de la célula, así como de los niveles de fluorescencia poblacional al variar el parámetro kse.

Parámetro ks1	Valor: 0.01				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	35 – 389	No	No	1.38621956
	1000	46 - 376	No	No	1.46155888
	1500	42 - 378	No	No	1.55917898
2000	24 - 441	No	No	1.66307884	

Parámetro ks1	Valor: 0.1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	17 – 444	No	No	13.9767784
	1000	32 - 432	Si	No	14.8444371
	1500	7 - 412	Si	No	15.7218869
2000	19 – 383	Si	No	16.954002	

Parámetro ks1	Valor: 1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	13 - 429	Si	Si	139.542048
	1000	20 - 413	Si	Si	148.671383
	1500	31 - 370	Si	Si	157.628711
2000	31- 403	Si	Si	169.092772	

Tabla 8. Comportamiento oscilatorio de las estrategias génicas y de los niveles de inducción interno y externo de la célula, así como de los niveles de fluorescencia poblacional al variar el parámetro ks1.

Parámetro ks0	Valor: 0.01				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	23 - 390	Si	Si	44.8242515
	1000	15 - 421	Si	Si	45.0872255
	1500	16 - 452	Si	Si	45.7425017
2000	22 - 416	Si	Si	45.9507944	

Parámetro ks0	Valor: 0.1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	22 - 410	Si	Si	42.8605908
	1000	14 - 423	Si	Si	43.4397026
	1500	27 - 395	Si	Si	44.3031883
2000	19 - 427	Si	Si	45.0998313	

Parámetro ks0	Valor: 1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	16 - 389	No	No	-8.33E+18
	1000	33 - 390	No	No	-8.46E+18
	1500	25 - 382	No	No	-8.54E+18
2000	51 - 425	No	No	-8.61E+18	

Tabla 9. Comportamiento oscilatorio de las estrategias génicas y de los niveles de inducción interno y externo de la célula, así como de los niveles de fluorescencia poblacional al variar el parámetro ks0.

Parámetro kdiff	Valor: 0.01				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	21 - 420	No	Si	40.2453653
	1000	13 - 422	No	Si	42.7195948
	1500	22 - 421	No	Si	45.3390579
2000	24 - 416	No	Si	48.3694291	

Parámetro kdiff	Valor: 0.1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	17 - 400	Si	Si	40.9218802
	1000	22 - 432	Si	Si	43.4776547
	1500	20 - 375	No	Si	46.0105043
2000	23 - 423	No	Si	49.2896088	

Parámetro kdiff	Valor: 1				
	Número de células	Rango de amplitud de las oscilaciones en las estrategias génicas	Oscilaciones del auto inductor externo	Oscilaciones de la fluorescencia poblacional	Valor promedio de inducción
	500	17 - 419	SI	SI	44.6477525
	1000	28 - 399	SI	SI	47.6873393
	1500	24 - 394	SI	SI	50.7136274
2000	24 - 422	SI	SI	54.7024311	

Tabla 10. Comportamiento oscilatorio de las estrategias génicas y de los niveles de inducción interno y externo de la célula, así como de los niveles de fluorescencia poblacional al variar el parámetro kdiff.

Estos parámetros analizados indican que la tasa de degradación y la tasa de difusión del autoinductor dentro y fuera de la célula afectan la inducción de las células al aumentar el autoinductor dentro y fuera de la célula. Las simulaciones llevadas a cabo muestran que este modelo resulta ser bastante sensible a los niveles de autoinductores. Esto es debido a que el modelo no considera varios factores celulares como son la división celular, la muerte celular ni el estado estacionario de la colonia bacteriana. Por ello el autoinductor no tiene forma de desaparecer una vez producido, por lo que si aumenta ligeramente su nivel tiene un efecto acumulativo en el aumento de la inducción de las células. Sin embargo este sistema de quorum sensing sí afecta al sistema genético del represilador, ya que los niveles de las oscilaciones de *luxI* fueron los más altos en las simulaciones. Esto genera un sistema de retroalimentación negativo inhibiendo la producción de *luxI* que a su vez genera la disminución de autoinductor interno (García-Ojalvo et al. 2004). Esto se ve claramente en la gráfica de las estrategias génicas verde comparado con su autoinducción.

Si bien observamos que al aumentar la densidad celular los niveles de inducción aumentan poblacionalmente ocasionando que la población alcance el umbral de inducción para producir fluorescencia es difícil determinar que observamos un fenómeno de sincronización. La sincronización es un sistema dinámico complejo no es un estado (Pikovsky *et al.*, 2001) por lo tanto en la dinámica de la fluorescencia no podría hablarse de sincronización ya que esta resulta más un estado estable de la fluorescencia debido a los niveles altos de la inducción. Otro punto importante sobre la carencia de sincronización a nivel de la fluorescencia es que la fluorescencia de por sí no es un fenómeno independiente del represilador ni de la actividad de las otras células. Una célula sin la actividad del represilador no podría mostrar la fluorescencia ya que el nivel de inducción (la concentración de autoinductor interno) estaría en niveles tan bajos que no alcanzaría el umbral para producir fluorescencia. Además precisamente como el sistema de percepción por quorum está regulado por la densidad celular no es posible generar oscilaciones de la fluorescencia si separamos la actividad fluorescente de cada célula como sistemas independientes. Por ello habría que considerar a la fluorescencia como un sólo sistema o un enorme oscilador. De tal forma que si queremos ver la sincronización subyacente a la fluorescencia debemos revisar el sistema del represilador de las células que son los verdaderos osciladores que se están intentando acoplar por medio de los autoinductores. Como se muestran en las gráficas de las estrategias génicas (Figura 22) es difícil discernir si las frecuencias de cada una de las oscilaciones concuerdan en los 6 juegos de estrategias génicas. Esta dificultad impide comprobar si hay un seguimiento de fase o de frecuencias, es decir, si $\Delta F = F_1 - F_2 = 0$ (Pikovsky *et al.*, 2001) para dos de las estrategias génicas. Por medio de transformaciones de Fourier es posible explorar las señales subyacentes del sistema y encontrar una frecuencia común entre la actividad de las estrategias génicas. De esta manera sería posible determinar si hay un acoplamiento de los osciladores y si esta se ve afectada también por los distintos parámetros que controlan la actividad de percepción por quorum.

Otro punto a observar es que Elowitz menciona en su trabajo que mientras más similar sea la actividad de los represiladores individuales en la población bacteriana, es más pequeño el umbral para alcanzar la sincronización. En este modelo la variabilidad de los sistemas del represilador quedan reducidos a 6, por ello el número de veces que fluoresce la población bacteriana, de manera sincronizada a lo largo de la simulación, es bastante frecuente. Otro aspecto importante sobre el control de la sincronización es la densidad poblacional. En general el sistema de quorum sensing se regula por medio de la densidad celular ya que es un medio de comunicación célula – célula (Waters & Bassler, 2005) y si la población es mayor hay mayor señalización hacia las células vecinas. Este modelo muestra que a mayor número de células hay mayor nivel de inducción y por lo tanto la mayoría de la población fluoresce, sin embargo los niveles de inducción no necesariamente son los mismos ya que los histogramas de inducción no muestran a la mayoría de las células en un mismo intervalo de inducción, más bien hay varias barras que se encuentran sobrepasando el nivel de inducción para presentar fluorescencia. Esta variabilidad es una dificultad propia del modelo ya que no hay un criterio fuertemente representativo para establecer el nivel de inducción al cual las células pueden presentar fluorescencia. Estas limitantes están presentes en cualquier modelo ya que solo se trata de una aproximación del sistema a estudiar, por lo tanto las conclusiones obtenidas no deben considerarse completamente verdaderas a la hora de evaluarlas en el sistema real. Más bien lo que resulta más útil es preguntarse si las diferencias entre el sistema y el modelo son lo suficientemente significativas para afectar cualquier conclusión derivada del modelo (Law & Kelton, 2003)

Otra limitación del modelo con respecto a la parte computacional y propia del programa es la ejecución de los comandos por parte de los agentes. Aunque el programa resulta bastante poderoso para realizar procedimientos y ejecutar comandos por parte de un gran número o prácticamente el total de agentes (ya sean patches o tortugas), esta capacidad se ve reducida si tenemos diferentes categorías de tortugas (breeds) o variables (turtles-own). En el desarrollo del modelo se encontró la dificultad de que algunos procedimientos, por ejemplo el conteo de los autoinductores, debía ser ejecutado desde la perspectiva de un solo agente ya que si se le pedía a un conjunto de agentes o a la población en general realizar el conteo, esto lo llevaba a cabo por cada uno de los agentes de la población, por lo que el valor final en el procedimiento no era la suma de autoinductores sino la suma de las sumas de autoinductores contados por cada una de las células. Estos y otros inconvenientes dificultaron la simplificación del código. También en la parte de la ejecución del procedimiento de piedra papel y tijeras, el código debió implementarse en un solo procedimiento extenso, ya que si el código se agrupaba en 3 diferentes procedimientos, más cortos que correspondían a cada una de las estrategias, había un sesgo a la hora de llevar a cabo los comandos ya que se ejecutaba el primer procedimiento sin tomar en cuenta las interacciones marcadas por los otros dos procedimientos subsecuentes.

También se vio que modelos evolutivos como la ecuación replicadora (Nowak, 2006) no podían ser implementados como tales en el código de Netlogo. Principalmente la aplicación de la matriz de pagos no generaba el comportamiento deseado. Si bien otros investigadores han realizado modelos aplicando esta ecuación replicadora para modelar estrategias con la dinámica de piedra papel y tijeras, las oscilaciones obtenidas no son muy estables y al parecer la composición de la población al final no tiene un efecto importante en el comportamiento de la dinámica poblacional (revisar referencias de las páginas web).

Es importante señalar que aunque no se han llevado a cabo experimentos *in vivo* del acoplamiento del represilador empleando el sistema de percepción por quorum, existen otras construcciones genéticas que si han podido sincronizar poblaciones bacterias empleando este sistema de comunicación celular. Por ejemplo los trabajos de Hasty y colaboradores en el 2010 y 2011 (Danino et. al 2010; Prindle et. al 2011) han logrado sincronizar colonias de bacterias empleando el sistema de percepción por quorum pero con construcciones genéticas que emplean sistemas de retroalimentación tanto positivo como negativo. Gracias a esto han podido diseñar microceldas donde se incuban las bacterias y dependiendo del flujo o de la difusión de las moléculas de sincronización, es posible sincronizar las colonias bacterianas para que presente fluorescencia a diferentes ritmos creando patrones de sincronización impresionantes como la ola mexicana o incluso la capacidad para funcionar como biopixeles en celdas fluorescentes.

9. Conclusiones

Por los resultados obtenidos se concluye que la densidad celular así como la difusión del autoinductor dentro y fuera de la célula, afecta el acoplamiento poblacional de manera positiva al aumentar los niveles de autoinducción de las células para generar fluorescencia. Las simulaciones indican que altas densidades poblacionales tienden a incrementar los niveles de inducción de las células y por lo tanto lo acercan al umbral de inducción para producir fluorescencia. Si bien a nivel genético genera un aumento en los niveles de las oscilaciones del mRNA de *lacl*, no se encontró una sincronización precisa de las oscilaciones en los 6 diferentes juegos de genes por lo tanto no se confirmó la hipótesis propuesta en el presente trabajo.

La actividad genética del represilador es posible implementarla en un modelo basado en agentes aplicando la teoría juegos a la dinámica de los 3 genes represores (*tetR*, *lacl* y *cl*) por medio del juego de piedra, papel y tijeras. Las oscilaciones del tipo sinusoidales de las estrategias génicas concuerdan con el modelo matemático de la actividad genética del mRNA. De esta manera, la posibilidad de trasladar netamente un modelo de ecuaciones diferenciales a un modelo con la capacidad de visualizar dicho fenómeno de manera espacial, es posible empleando modelos basados en agentes.

El presente trabajo no solo concluye aquí, los resultados abren la posibilidad de explorar el fenómeno de sincronización empleando otras herramientas tanto matemáticas como computacionales. Por ejemplo: el análisis de los datos de las oscilaciones por medio de una transformación de Fourier permitiría analizar mejor los patrones de sincronización de las oscilaciones al observar las frecuencias y determinar si estas generan una sincronización en fase o antifase. El análisis paramétrico de todas las constantes también ayudaría a dilucidar mejor el papel de las reacciones de cada componente en la dinámica molecular. Si bien no se sabe con precisión los valores reales de degradación y síntesis tanto del mRNA como de las proteínas, tener un modelo que nos permita explorar diferentes parámetros permitirá predecir mejor el comportamiento de la dinámica tanto de los genes como de las proteínas.

La implementación en otras plataformas de modelos basados en agentes también es una opción para validar este modelo y explorar el potencial de los modelos basados en agentes. Muchos trabajos se han llevado cabo en el campo de la ecología y cualquier área de estudio que involucre poblaciones. Sin embargo los modelos basados en agentes tienen el potencial de emplearse en cualquier área de estudio, ya sea que involucre poblaciones o un solo individuo. La facilidad del lenguaje y de la interfaz para desarrollar cualquier modelo, sugiere que este tipo de herramientas computacionales deberían ser una base primordial, tanto en la enseñanza como en la investigación si se quiere tener una visión y observación más enriquecedora, con lo cual podamos abordar los nuevos paradigmas que arroja la naturaleza.

También más recientemente herramientas computacionales como matemática, paquetes estadísticos como R y muchos otros software de diseño basados en java (Netlogo, VCell) y en C++ (BioNetGen) permiten crear modelos para estudiar sistemas dinámicos en biología (de lo que se

encarga la biología de sistemas) por lo que también se podrían emplear estas y muchas otras herramientas en el estudio de los ritmos biológicos y la sincronización.

Por otro lado estas simulaciones computacionales sugieren que implementar esta construcción genética *in vivo* dentro del laboratorio generaría datos interesantes para comprender aún mejor el fenómeno colectivo de sincronización en bacterias. En especial si la densidad cumple el papel crítico en la sincronización, en cuyo caso describiría muy bien la situación que se presenta, por ejemplo, en un quimiostato. O si depende más bien de la difusión del autoinductor extracelular hacia las células, por lo que hablaríamos de los sistemas diseñados por Hasty y colaboradores (Danino *et al.* 2010) de dispositivos de microceldas acrílicas donde sería posible controlar el flujo de autoinductor y por lo tanto controlar el flujo de información que es transmitido a las células por el sistema de percepción de quorum.

Además tener un modelo que incorpore tantas variables y parámetros que definen a un sistema caótico y observar la interacción de estos componentes y que obtengamos resultados que podamos analizar, considero que es un gran avance para el desarrollo de la ciencia. La biología sintética ha dado ese primer intento de unificar varias áreas del conocimiento científico (física, computación, matemáticas y biología) para comprender fenómenos que anteriormente la biología no podía comprender por sí sola. El camino para el entendimiento de comportamientos emergentes a partir del estudio de sus componentes individuales está iluminado por la modelación matemática o computacional y sobre todo en biología, que hablamos de sistemas increíblemente complejos (pero no por ello menos asombrosos e interesantes) debe ser la nueva panacea de la investigación.

El fenómeno de sincronización que se observa a diferentes niveles de organización desde el celular hasta el poblacional contiene los mismos componentes que influyen en sistemas físicos como los lasers o los relojes de péndulo: oscilaciones que se ajustan a una escala de tiempo determinada. Esto nos habla de que es posible entender los fenómenos de sincronización biológica aplicando los mismos criterios de estudio que se aplican a objetos mecánicos y físicos. El problema radica en que no entendemos tan bien cómo funcionan los sistemas biológicos y más aún a nivel molecular. Por ello el diseño de construcciones biológicas sintéticas también arroja luz al esclarecimiento de estos problemas de los sistemas complejos con propiedades emergentes. La biología sintética tiene el potencial de ser una nueva área que se encargue de romper los viejos paradigmas de investigación en biología al conjuntar las matemáticas, física y computación con la biología para el diseño de construcciones moleculares sintéticas. La biología de sistemas también va de la mano y nuevas áreas como la biología cuántica contribuirán al entendimiento de algunas de las interrogantes que han persistido en biología desde el inicio de la ciencia. ¿Qué es lo que determina que algo este vivo? ¿Cómo surgió o puede surgir la vida? ¿Podemos modificar a un organismo a tal grado que nosotros cumplamos el papel de una fuerza evolutiva?

Referencias

1. Adamatzky, A. *Game of Life Cellular Automata*. 2010. Primera edición. Springer-Verlag. Londres. UK. Pp. 1-33
2. Albrecht, U. *The Circadian Clock*. 2010. Primera edición. Springer. USA. pp. 1-12.
3. Allada, R; Emery, P; Takahashi, J; Rosbash, M. STOPPING TIME: The Genetics of Fly and Mouse Circadian Clocks. 2001. *Annu. Rev. Neurosci.* 24:1091–119
4. Atkinson, MR; Savageau, MA; Myers, JT; Ninfa ,AJ: Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. 2003. *Cell.* 113:597-607.
5. Auel, D; Fussenegger, M. Watch the clock—engineering biological systems to be on time. *Current Opinion in Genetics & Development* 2010, 20:634–643.
6. Bass, J. Circadian topology of metabolism.2012. *Nature* 491, 348–356.
7. Binmore, K. *Game Theory: A Very Short Introduction*. 2007. Primera edición. Oxford University Press Inc., New York. USA. pp. 117-139
8. Broom, M; Rychtar, J. *Game-Theoretical Models in Biology*. 2013. Primera edición. CRC Press Taylor & Francis Group. 4-10, 71-80
9. Danino, T; Mondragón-Palomino, O; Tsimring, L; Hasty, J. A synchronized quorum of genetic clocks. *Nature*. Vol 463. 21 Enero 2010.
10. Deutsch, A; Dormann, S. *Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis*. Primera edición. 2005. Birkhäuser. USA. pp. 59- 78.
11. Dubitzky, W., Wolkenhauer, O., Yokota, H., Cho, K. *Encyclopedia of Systems Biology*. 2013. Springer References. Primera edición. pp. 843-844
12. Dunlap, J; Loros, J; DeCoursey, P. *Chronobiology Biological Timekeeping*. 2004. Primera edición. Sinauer Associates, Inc. USA. Pp.
13. Ermentrout, G; Edelstein-Keshet, L. *Cellular Automata Approaches to Biological Modelling*. *J. theor. Biol.* (1993) 160, 97-133.
14. Elowitz, M, Leibler, S. A synthetic oscillatory network of transcriptional regulators. 2000. *Nature* 403, 335-33.
15. Frean, M; Abraham, E. Rock-scissors-paper and the survival of the weakest. 2001. *Proc Biol Sci.*; 268(1474): 1323–1327
16. Garcia-Ojalvo, J; Elowitz, M; Strogatz, S. Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing.2004. *PNAS.* 101: 10955-10960
17. Goldbeter, A. *Biochemical oscillations and cellular rhythms: the molecular bases of periodic and chaotic behavior*. 1996. Primera edición. Cambridge University Press. UK. pp. 1-27
18. Goldbeter, A. Computational approaches to cellular rhythms. 2002. *Nature* 420, 238-245.
19. Gorochoowski, T; Matyjaszkiewicz, A; Todd, T; Oak, N; Kowalska, K et al. BSim: An Agent-Based Tool for Modeling Bacterial Populations in Systems and Synthetic Biology. 2012. *PLOS ONE*. Volume 7. Issue 8. e42790.
20. Grimm, V; Berger, U; DeAngelis, D; Polhill, J; Giske, J; Railsback, S. The ODD protocol: A review and first update. 2010. *Ecological Modelling* 221.2760–2768

21. Hardin, P. E., Hall, J. C. & Rosbash, M. Feedback of the *Drosophila* period gene product on circadian cycling of its messenger RNA levels. 1990. *Nature* 343, 536–540.
22. Ingalls, B. *Mathematical Modelling in Systems Biology An Introduction*. 2013. Primera edición. MIT Press. USA. pp. 253-255.
23. Koukkari, w; Sothorn, R. *Introducing biological rhythms*. 2006. Primera edición. Springer. USA. pp. 1-34
24. Konopka, R.J. & Benzer, S. Clock mutants of *Drosophila melanogaster*. 1971. *Proc. Natl Acad. Sci. USA* 68, 2112–2116.
25. Lema, M., Echave, J.n., and Golombek, D.A. Too many mathematical models of circadian clocks. 2001. *Biol. Rhythm Res.* 32, 285–298.
26. Lewontin, R. *Evolution and the Theory of Games*. 1961. *J. Theoret. Biol.* 1, 382-403
27. Lorek, H; Sonnenschein, M. Modelling and simulation software to support individual-based ecological modelling. 1999. *Ecological Modelling.* 115. 199–216
28. Macal, C.; North, M. Tutorial on agent-based modeling and simulation. 2005. In *Proceedings of the 37th conference on Winter simulation* (pp. 2-15). Winter Simulation Conference.
29. Macal, C; North, M. Tutorial on agent-based modelling and simulation. *Journal of Simulation*. 2010 4, 151–162
30. Mads F., Madsen; Sune, Dano; Preben G. Sorensen. On the mechanisms of glycolytic oscillations in yeast. 2005. *FEBS Journal* 272. 2648–2660
31. McMillen, D., Kopell, N., Hasty, J. & Collins, J. J. 2002. *Proc. Natl. Acad. Sci. USA* 99, 679–684.
32. Mishchenko, Y. Oscillations in Rational Economies. 2014. *PLoS ONE*. Volume 9. Issue 2.
33. Nowak, M. *Evolutionary dynamics: exploring the equations of life*. 2006. Primera edición. The Belknap Press of Harvard University Press. Canada. pp. 45-63
34. Pikovsky, A; Rosenblum, M; Kurths, J. *Synchronization A universal concept in nonlinear sciences*. 2001. Primera edición. Cambridge University Press. UK. pp. 1 -134
35. Otto, S; Day, T. *A biologist's guide to mathematical modeling in ecology and evolution*. 2007. Primera edición. Princeton University Press. USA. pp. 17 -50.
36. R.A. Wever, *The Circadian System of Man, Results of Experiments under Temporal Isolation*. 1979. Springer-Verlag. New York. 276 pp.
37. Railsback, S; Grimm, V. *A Course in Individual-based and Agent-based Modeling*. 2009. Primera edición. Princeton University Press. USA.
38. Reading, N; Sperandio, V. Quorum sensing: the many languages of bacteria. 2006. *FEMS Microbiol. Lett.* 254, 1–11 .
39. Rice, S. *Evolutionary Theory Mathematical and Conceptual Foundations*. 2004. Primera edición. Sinauer Associates, Inc. Publishers. USA. pp. 263-289
40. Roenneberg, T; Chua, E; Bernardo, R; Mendoza, E. Modelling Biological Rhythms. 2008. *Current Biology* 18, R826–R835.
41. Rosenberg, E ; DeLong, E; Lory, S; Stackebrandt, E; Thompson, F. *The Prokaryotes Prokaryotic Communities and Ecophysiology*. 2013. Cuarta edición. Springer Reference. Pp. 495-506.

42. Sehgal, A.; Price, J.L.; Man, B. ; Young, M.W. Loss of circadian behavioral rhythms and per RNA oscillations in the *Drosophila* mutant timeless. 1994. *Science* 263, 1603–1606.
43. Sinervo B, Lively, C. The rock-paper-scissors game and the evolution of alternative male strategies. 1996. *Nature*. Vol 380.
44. Singh, V. Recent advancements in synthetic biology: Current status and challenges. 2013. *Gene* .535. 1–11.
45. Sklar, E. Software Review: NetLogo, a Multi-agent Simulation Environment.2007. *Artificial Life* 13: 303–311
46. Smith, T; Smith, R. *Ecología*. 2007. Sexta edición. Pearson Educación. Mexico. pp. 299-320.
47. Smolensky, M; Peppas, N. Chronobiology, drug delivery, and chronotherapeutics. 2007. *Advanced Drug Delivery Reviews* 59. 828–851.
48. Strogatz, S. *Sync: how order emerges from chaos in the universe, nature, and daily life*. 2003. Primera edición. Hyperion books. USA. pp 11-289
49. Strogatz, S; Stewart, I. Coupled Oscillators and Biological Synchronization A subtle mathematical thread connects clocks, ambling elephants, brain rhythms and the onset of chaos.1993. *Scientific American*.269, 102-109.
50. Wager-Smith, K; Kay, S. Circadian rhythm genetics: from flies to mice to humans.2000. *Nature Genetics*. 26, 23– 27.
51. Ward, J; King, J; Koerber, A; Williams, P; Croft, J; Socket,R. Mathematical modelling of quorum sensing in bacteria. 2001. *Math Med Biol*. 18 (3). 263-292
52. Weibel, P. *Beyond Art: A Third Culture: A Comparative Study in Cultures, Art and Science in 20th Century Austria and Hungary*. 2005. Springer. Primera edición. USA. pp. 368-416.
53. Wilensky, U. Modeling nature’s emergent patterns with multi-agent languages. 2002. *Proceedings of EuroLogo 2002*. Linz, Austria
54. Wolfram, S. Cellular automata as models of complexity. 1984. *Nature*. **311**. 419 - 424
55. Young, M. W. & Kay, S. Time zones: a comparative genetics of circadian clocks.2001. *Nature Rev. Genet*. 2, 702–715.

Referencias electrónicas

1. Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/>. Recuperado el 24 de Noviembre del 2014
2. Bennett, N. NetLogo Tutorial Series: Rock-Paper-Scissors Ecosystem. Recuperado el 12 de Julio del 2015. <http://origins.santafe.edu/system/files/Rock-Paper-Scissors%20Ecosystem.pdf>.
3. Repressilator from the Wolfram Demonstrations Project. Recuperado el 19 de Marzo del 2015. <http://demonstrations.wolfram.com/Repressilator/>.
4. Complexity Explorer. Recuperado el 16 febrero del 2015. <http://www.complexityexplorer.org/>

5. Stack Overflow. Recuperado el 19 de Noviembre del 2015. <http://stackoverflow.com/>
6. Laboratorio de Aprendizaje de NetLogo. Recuperado el 12 de Marzo del 2015. <http://online.sfsu.edu/jjohnson/NetlogoTranslation/index.html>
7. Welcome to the Modeling Commons! NetLogo Modeling Commons. Recuperado el 26 de Mayo del 2015. <http://modelingcommons.org/account/login>

Apéndice A: Código en Netlogo

Código del modelo poblacional en Netlogo

El código se divide en tres partes: el establecimiento de las variables y sus propiedades, el procedimiento para la creación del mundo y las tortugas (setup) y el procedimiento que contiene las instrucciones para que cada tortuga y elemento del mundo realicen ciertas acciones de interacción y movimiento(go)

Variables

Estos procedimientos definen el tipo de raza de (breed) de cada población de tortugas así como las variables propias de cada raza. También define una variable global a la cual todos los agentes tienen acceso.

```
globals  
[  
  colors  
]  
breed [cells cell]  
breed [autoinducers autoinducer]  
cells-own [induccion]
```

Creación del mundo y las tortugas

Los siguientes procedimientos establecen la cantidad de tortugas según su raza y la manera en que queda delimitado el mundo. Los procedimientos “create-genes” con numeración del 1 al 3 requieren la creación de un slider en la interfaz nombrada “genes” con valor del 0 al 1000 y con incremento de 10. El procedimiento “create-celulas” también requiere la creación de un slider nombrándolo “Número-celulas” con un valor de 0 a 3000 con un incremento de 100.

```
to setup  
  clear-all
```

```
create-celulas
create-genes1
create-genes1-2
create-genes2
create-genes2-2
create-genes3
create-genes3-3
create-AI
reset-ticks
end
```

```
to create-genes1
  set colors [red green blue]
  create-turtles genes
  [
    set color item (who mod 3) (colors)
    set shape "circle 2"
    set xcor (-40 + random 25 )
    set ycor (-13 + random -13)]
  ask patches with[pxcor = -14 and pycor < -12]
  [set pcolor white]
end
```

```
to create-genes1-2
  set colors [red green blue]
  create-turtles genes
  [
    set color item (who mod 3) (colors)
    set shape "circle 2"
    set xcor (-40 + random 25 )
    set ycor (-40 + random 12)]
end
```

```
to create-genes2
  set colors [red green blue]
  create-turtles genes
  [
    set color item (who mod 3) (colors)
    set shape "circle 2"
    set xcor (-12 + random 25 )
```

```
set ycor (-13 + random -13])
ask patches with[pxcor = 14 and pycor < -12]
[set pcolor white]
end
```

```
to create-genes2-2
  set colors [red green blue]
  create-turtles genes
  [
    set color item (who mod 3) (colors)
    set shape "circle 2"
    set xcor (-12 + random 25 )
    set ycor (-40 + random 12)]
end
```

```
to create-genes3
  set colors [red green blue]
  create-turtles genes
  [
    set color item (who mod 3) (colors)
    set shape "circle 2"
    set xcor (40 - random 25 )
    set ycor (-13 + random -13)]
end
```

```
to create-genes3-3
  set colors [red green blue]
  create-turtles genes
  [
    set color item (who mod 3) (colors)
    set shape "circle 2"
    set xcor (40 - random 25 )
    set ycor (-40 + random 12)]
end
```

```
to create-celulas
  set-default-shape cells "cell"
  create-cells Número-celulas
  [
    set color gray
```

```

    set size 5
    set xcor (35 + random -70 )
    set ycor (36 + random -44)]
ask patches with[pycor = -12 ]
[set pcolor white]
ask patches with[pycor = -27 ]
[set pcolor white]
end

```

```

to create-AI
    set-default-shape autoinducers "ai"
end

```

Interacción y movimiento

Los siguientes procedimientos permiten la movilidad de las tortugas así como la interacción entre estas y el mundo y todo lo relacionado con la dinámica explicada en la metodología. Para llevar a cabo estas acciones se crean 2 botones: el botón “setup” que corresponde a los procedimientos del apartado anterior y el botón “go” que se describirá a continuación y que puede seleccionarse la opción de observación continua. El botón “go” realiza 11 diferentes procedimientos en cada corrida. Para que estos procedimientos puedan operar es necesario crear 7 diferentes botones. Para el procedimiento “display-labels” es necesario crear un boton tipo “interruptor” y nombrarlo como “Ver-induccion?”. Para los procedimientos “aleatory-difusion-in”, “difusión-out” y “lac-recali-...” del 1 al 3, es necesario crear 6 sliders. 5 sliders se nombraran como: “kdiff”, “kse”, “ks0”, “ks1” y “n” respectivamente. Estos sliders tendrán un valor de 0 a 1 con un incremento de 0.1. El sexto slider se nombrara “k” pero su valor ira de 0 a 20 con un incremento de 0.1.

```

to go
ask cells [ bounce fd 1 ]
ask autoinducers [bounce fd 1]
ask turtles with [color = red] [bounce fd 1]
ask turtles with [color = green] [bounce fd 1]
ask turtles with [color = blue] [bounce fd 1]
play
display-labels
aleatory-difusion-in
ask one-of patches with [pycor > 25] [difusion-out]
flourescence

```

```

lac-recali-1
lac-recali-1-1
lac-recali-2
lac-recali-2-2
lac-recali-3
lac-recali-3-3
if ticks = 500 [stop]
tick
end

```

```

to flourescence
ask cells
[
if induccion > 50
[set color yellow]
if induccion < 50
[set color gray]
]
end

```

```

to lac-recali-1
let ajuste1 [induccion] of one-of cells
if ajuste1 != -1
[let calibrar1 ((k * ajuste1) / (1 + ajuste1))
if calibrar1 < count turtles with [ xcor < -13 and ycor < -12 and ycor > -26 and color != green] and
calibrar1 > 0
[
ask n-of calibrar1 turtles with [ xcor < -13 and ycor < -12 and ycor > -26 and color != green]
[die]
ask one-of turtles with [color = green and xcor < -13 and ycor < -12 and ycor > -27]
[hatch calibrar1
[set color green
set shape "circle 2"
set xcor (-40 + random 25 )
set ycor (-13 + random -13)]
]
]
]

```

```
end
```

```
to lac-recali-1-1
```

```
  let ajuste1 [induccion] of one-of cells
```

```
  if ajuste1 != -1
```

```
  [
```

```
    let calibrar1 ((k * ajuste1) / (1 + ajuste1))
```

```
    if calibrar1 < count turtles with [ xcor < -13 and ycor < -27 and color != green] and calibrar1 > 0
```

```
    [
```

```
ask n-of calibrar1 turtles with [ xcor < -13 and ycor < -27 and color != green]
```

```
[die]
```

```
ask one-of turtles with [color = green and xcor < -13 and ycor < -27 ]
```

```
[hatch calibrar1
```

```
  [set color green
```

```
    set shape "circle 2"
```

```
    set xcor (-40 + random 25 )
```

```
set ycor (-40 + random 12)]
```

```
]
```

```
]
```

```
]
```

```
end
```

```
to lac-recali-2
```

```
  let ajuste2 [induccion] of one-of cells
```

```
  if ajuste2 != -1
```

```
  [
```

```
    let calibrar2 ((k * ajuste2) / (1 + ajuste2))
```

```
    if calibrar2 < count turtles with [ xcor > -12 and xcor < 11 and ycor < -12 and ycor > -26 and color != green] and calibrar2 > 0
```

```
    [
```

```
ask n-of calibrar2 turtles with [ xcor > -12 and xcor < 11 and ycor < -12 and ycor > -26 and color != green]
```

```
[die]
```

```
ask one-of turtles with [color = green and xcor > -12 and xcor < 11 and ycor > -26 and ycor < -12]
```

```
[hatch calibrar2
```

```
  [set color green
```

```
    set shape "circle 2"
```

```
    set xcor (-12 + random 23 )
```

```

    set ycor (-13 + random -13) ]
  ]
]
end

```

```

to lac-recali-2-2
  let ajuste2 [induccion] of one-of cells
  if ajuste2 != -1
  [
    let calibrar2 ((k * ajuste2) / (1 + ajuste2))
    if calibrar2 < count turtles with [ xcor > -12 and xcor < 11 and ycor < -26 and color != green] and
    calibrar2 > 0
    [
      ask n-of calibrar2 turtles with [ xcor > -12 and xcor < 11 and ycor < -26 and color != green]
    [die]
    ask one-of turtles with [color = green and xcor > -12 and xcor < 11 and ycor < -26]
    [hatch calibrar2
      [set color green
        set shape "circle 2"
        set xcor (-12 + random 25)
        set ycor (-40 + random 12) ]
    ]
  ]
end

```

```

to lac-recali-3
  let ajuste3 [induccion]of one-of cells
  if ajuste3 != -1
  [
    let calibrar3 ((k * ajuste3) / (1 + ajuste3))
    if calibrar3 < count turtles with [ xcor > 11 and ycor < -12 and ycor > -26 and color != green] and
    calibrar3 > 0
    [
      ask n-of calibrar3 turtles with [ xcor > 11 and ycor < -12 and ycor > -26 and color != green]
    [die]
    ask one-of turtles with [color = green and xcor > 11 and ycor < -12 and ycor > -26]
    [hatch calibrar3
      [set color red

```

```

    set shape "circle 2"
    set xcor (40 - random 28 )
    set ycor (-13 + random -13) ]
  ]
]
end

```

```

to lac-recali-3-3
  let ajuste3 [induccion] of one-of cells
  if ajuste3 != -1
    [
    let calibrar3 ((k * ajuste3) / (1 + ajuste3))
    if calibrar3 < count turtles with [ xcor > 11 and ycor < -26 and color != green] and calibrar3 > 0
    [
    ask n-of calibrar3 turtles with [ xcor > 11 and ycor < -26 and color != green]
    [die]
    ask one-of turtles with [color = green and xcor > 11 and ycor < -26]
    [hatch calibrar3
    [set color green
    set shape "circle 2"
    set xcor (40 - random 28 )
    set ycor (-40 + random 12) ]
    ]
    ]
    ]
end

```

```

to aleatory-difusion-in
  ask one-of cells
  [
  let Ai1 count turtles with [color = red and xcor < -13 and ycor > -26]
  let Ai2 count turtles with [color = red and xcor < -13 and ycor < -27]
  let Ai3 count turtles with [color = red and xcor > -12 and xcor < 11 and ycor > -26]
  let Ai4 count turtles with [color = red and xcor > -12 and xcor < 11 and ycor < -26]
  let Ai5 count turtles with [color = red and xcor > 11 and ycor > -26]
  let Ai6 count turtles with [color = red and xcor > 11 and ycor < -26]
  let LAi1 list Ai1 Ai2

```

```

let LAi2 list Ai4 Ai5
let LA sentence LAi1 Ai3
let LB sentence LAi2 Ai6
let LC sentence LA LB
let Se1 count autoinducers
ask cells
[
let menus1 5
let Si1 [induccion] of self + ((sum [induccion] of cells in-radius 1) / menus1)
let concentracion-si1 (((- (ks0 * Si1)) + ( ks1 * one-of LC ) - (n * (Si1 - Se1))))
let incremento-si1 (((- (ks0 * concentracion-si1)) + (ks1 * one-of LC ) - (n * (concentracion-si1 -
Se1))))
set induccion round incremento-si1
]
]
end

```

```

to difusion-out
let promedio-si (sum [induccion] of cells / count cells)
let Se-e count autoinducers
let concentracion-se ((- kse * Se-e) + (kdiff * (promedio-si - Se-e)))
let incremento-se ((- kse * concentracion-se) + ( kdiff * (promedio-si - Se-e )))
If incremento-se > 0
[
ask autoinducers
[die]
]
sprout-autoinducers incremento-se
[
set shape "ai"
set color yellow
set size 1
set xcor (38 + random-float -76 )
set ycor (38 + random-float -49)
]
end

```

```

to display-labels
ask cells [ set label "" ]

```

```
if ver-induccion? [  
  ask cells [ set label round induccion ]  
]  
end
```

```
to play  
  ask turtles [  
    let mycolor color  
    let nearby other turtles in-radius 1  
    if mycolor = blue [  
      if any? nearby with [color = green]  
      [  
        set color blue  
      ]  
      if any? nearby with [color = red]  
      [  
        set color red  
      ]  
    ]  
    if mycolor = green [  
      if any? nearby with [color = red]  
      [  
        set color green  
      ]  
      if any? nearby with [color = blue]  
      [  
        set color blue  
      ]  
    ]  
    if mycolor = red [  
      if any? nearby with [color = blue]  
      [  
        set color red  
      ]  
      if any? nearby with [color = green]  
      [  
        set color green  
      ]  
    ]  
  ]  
end
```

```
]
end
```

```
to bounce
  if abs pxcor = max-pxcor
    [ set heading (- heading) ]
  ;; bounce off top and bottom walls
  if abs pycor = max-pycor
    [ set heading (180 - heading) ]
    if [pcolor] of patch-at dx 0 = white [
      set heading (- heading)
    ]
    if [pcolor] of patch-at 0 dy = white [
      set heading (180 - heading)
    ]
  ]
end.
```

Apéndice B: Código en Polymath

Interfaz de Polymath

Polymath es un programa de resolución de ecuaciones diferenciales que hace uso de una interfaz para desarrollar las ecuaciones y contiene algoritmos para resolver sistemas lineales de ecuaciones, sistemas de ecuaciones no lineales y sistemas de ecuaciones diferenciales. Los resultados pueden ser visualizados en graficas o en tablas de datos. La sintaxis de las ecuaciones es similar a como se desarrolla en otros programas de resolución de ecuaciones como Mathematica o Matlab. Se establecen las ecuaciones, luego se determinan los valores de las constantes (si es que hay) y de las variables en el tiempo 0, y se delimita el tiempo o cualquier otra variable independiente. El código empleado para resolver las ecuaciones de Elowitz, de su modelo matemático y comparar de forma cualitativa estos resultados con el modelo de Netlogo es el siguiente:

Sistema de ecuaciones diferenciales

```
d(ai)/d(t) = - (ai) + ((216)/(1+(Ci)^2))
d(Ai)/d(t) = - (10 * (Ai - ai))
d(bi)/d(t) = - (bi) + ((216)/(1+(Ai)^2))
```

$$d(Bi)/d(t) = - (10 * (Bi - bi))$$

$$d(ci)/d(t) = - (ci) + ((216)/(1+(Bi)^2)) + (((k * (Si)) / (1 + (Si))))$$

$$d(Ci)/d(t) = - (10 * (Ci - ci))$$

$$d(Si)/d(t) = - (ks0 * Si) + (ks1 * Ai) - (n * (Si - Se))$$

$$d(Se)/d(t) = - (kse * Se) + (kdiff * ((Si / 10) - Se))$$

Valores de las constantes y de las variables dependientes en el tiempo 0

$$ai(0) = 20$$

$$bi(0) = 20$$

$$ci(0) = 20$$

$$Ai(0) = 20$$

$$Bi(0) = 20$$

$$Ci(0) = 30$$

$$Si(0) = 25$$

$$Se(0) = 25$$

$$k = 0.000005$$

$$ks0 = 0.2$$

$$ks1 = 0.2$$

$$n = 0.2$$

$$kse = 0.2$$

$$kdiff = 0.9$$

Delimitación del tiempo o de la variable independiente

$$t(0) = 0$$

$$t(f) = 150$$