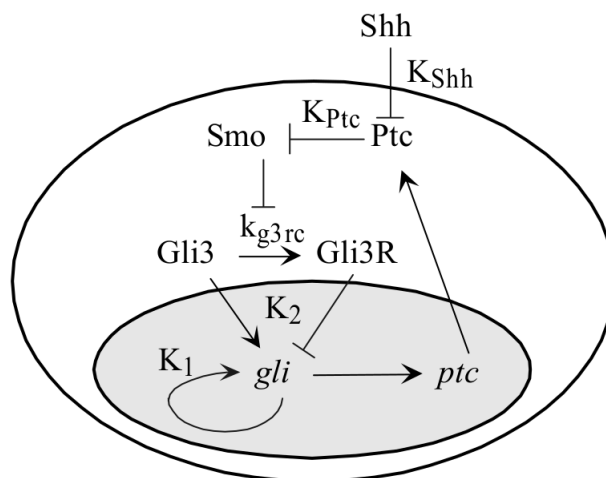




Universidad de Granada
Máster en Física y Matemática
Departamento de Matemática Aplicada

Nuevo modelo matemático para el sistema de señalización de la proteína Sonic Hedgehog (Shh)

Bartolomé Ortiz Viso
Tutor: Óscar Sánchez



Septiembre 2018

DECLARACIÓN

En cumplimiento de la normativa aprobada en Consejo de Gobierno de 4 de marzo de 2013, sobre Directrices de la Universidad de Granada para el desarrollo de la asignatura "Trabajo Fin de Máster" de sus títulos de máster (Art 8,4)

D.D^aBartolomé Ortiz Viso.....

Asume la originalidad del trabajo fin de máster, entendida en el sentido de que no ha utilizado fuentes sin citarlas debidamente.

Granada, a 03 de Septiembre de 2018.

Fdo.:



Abstract

During human development, cells are exposed to a complex network of regulatory signals, which must be interpreted correctly in order to success doing functions necessary for the organism. Therefore, the transition of signals and cascades of genetic regulation can be understood as mechanisms of information processing that translate extracellular information into intracellular decisions.

The present work aims to show the differences, in terms of a qualitative behavior, that can be found through these complex systems of biological regulation process through different theoretical approaches.

We were particularly interested in the Shh signaling system, among its many roles during development, patterns spinal cord and limb bud tissue differentiation and controls midbrain and ventral forebrain neuronal differentiation.

There are a few models that has been described in order to understand its behaviour. The most important was developed in [Lai et al., 2004]. In particular, these mathematicians use the thermodynamic approach to Shh's gene expression mechanish. This approach, in rough outlines, aim to model the gene transcription systems enumerating all the possible states of the promoter and enchancers of gene transcription activation, and then, relating all of them with their theoretically calculate transcriptional activation level. While these steps can be done in multiple ways, they focused their work on the stimulated approach, which links the transcriptional activity to the transcriptional factors (uniquely).

However, some discrepancies has been observed in biological experiments, mainly focused in the existence of an unique stable state, casting aside the biestable swicht behavior shown in the classic model.

As a result of these state of the art, we thought it would be necessary to offer a new model that update the classic one, but at the same time, we still want to use the BEWARE approach.

Our main goal has been to develop and study a new model based on the same BEWARE strategy but approaching it by the recruitment perspective, that is, taking into account the fundamental part of the RNAP in this whole biological process and its hability of produce the transcriptional activation.

This work presents a deduction of our new model based on the approach made in [Cam-bon and Sanchez, 2017] and a qualitative study of both models (old and new) through numerical simulations (numerical integration, parameter tuning, bifurcation diagrams, steady states, etc), our discoveries during these and our conclusions about the new and old models.

Specifically, we linked the biestable behaviour of the old model not only to the Shh/K_{Shh} ratio in our cell but with the basal rate of Gli3 (a common trancription factor in this process). We haven't seen that last link in any paper.

Furthermore, we found that our new model offers a surprising searched conclusion. Even though the main thermodynamic operator exhibit a highly similar behaviour, our global simulations show a single steady state, no matter how far we alter our parameters.

We hope that these experiments should motivate a deep analytic research of our new model, because our results suggest it could fit the actual biological paradigm, helping us to understand quite a lot about this important topic.

Key Words: *Signaling Systems, Sonic Hedgehog, Math Models, Beware models*

Agradecimientos

I would like to express my gratitude to:

- My sister and my parents. Thank you for being there every time I need you.
- My supervisor: Óscar Sánchez.

'Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful'.

George Box, 1987.

Índice general

| | |
|---|-----------|
| Abstract | II |
| Agradecimientos | v |
| 1. Introducción | 1 |
| 1.1. Motivación y objetivos | 1 |
| 1.2. Sistema de señalización de Shh | 3 |
| 1.2.1. Descripción bioquímica del proceso | 3 |
| 1.2.2. Interacción entre Ptc y Shh | 6 |
| 1.2.3. Señal de transcripción | 7 |
| 1.2.4. Dinámica de Gli3 y Gli3R | 8 |
| 1.3. Modelado BEWARE | 9 |
| 2. Modelo clásico | 15 |
| 2.1. Introducción | 15 |
| 2.2. Modelado BEWARE | 15 |
| 2.3. Sistema final | 17 |
| 2.4. Estados estacionarios | 18 |
| 2.5. Simulaciones | 20 |

| | |
|---|-----------|
| 2.5.1. Parámetros | 20 |
| 2.5.2. Variación del operador BEWARE (promoter y basal) | 20 |
| 2.5.3. Evolución temporal | 22 |
| 2.5.4. Análisis numérico de los estados estacionarios | 24 |
| 2.5.5. Diagramas de bifurcación | 25 |
| 2.6. Críticas | 29 |
| 3. Modelo alternativo | 31 |
| 3.1. Introducción | 31 |
| 3.2. Modelado BEWARE | 32 |
| 3.2.1. Cálculo del operador | 33 |
| 3.3. Sistema final | 39 |
| 3.4. Estados estacionarios | 40 |
| 3.5. Simulaciones | 41 |
| 3.5.1. Parámetros | 41 |
| 3.5.2. Variación del operador BEWARE | 41 |
| 3.5.3. Evolución temporal | 42 |
| 3.5.4. Análisis numérico de los estados estacionarios | 43 |
| 4. Conclusiones | 47 |
| 4.1. Conclusiones y trabajo futuro | 47 |
| Bibliografía | 48 |

| | |
|-----------------------|-----------|
| A. Códigos | 52 |
| A.1. Python | 52 |
| A.2. Octave | 83 |
| A.3. XppAut | 84 |

Índice de cuadros

| | |
|--|----|
| 2.1. Tabla de parámetros de [Lai et al., 2004] | 21 |
| 3.1. Tabla de parámetros, operador <i>BEWARE</i> | 42 |

Índice de figuras

| | |
|--|----|
| 1.1. Representación esquemática de la proteína de Shh. Fuente: [Wikipedia contributors, 2018b] | 2 |
| 1.2. Representación esquemática de la proteínas Gli1, Gli3 y Smo. Fuente: [Hornbeck PV, 2015] | 4 |
| 1.3. Representación esquemática de la red de transcripción. Fuente: [Lai et al., 2004] | 5 |
| 1.4. Representación artística de la transcripción génica, [Wikipedia contributors, 2018c]. Traducción y modificación propia | 11 |
| 1.5. 1: ARN polimerasa, 2: represor, 3: promotor, 4: operador, 5: inhibidor del represor, 6-8: Genes. Arriba: el gen está esencialmente desactivado. Abajo: el gen está encendido [Wikipedia contributors, 2018a]. Traducción y modificación propia. | 12 |
| 2.1. Variación del Operador Promoter bajo la variación de Gli3R | 22 |
| 2.2. Variación del Operador Basal bajo la variación de Gli3R | 23 |
| 2.3. Comparación numérica de las formas de reducidas de Promoter y Basal (en negro discontinuo) frente a las formas de [Lai et al., 2004] | 24 |
| 2.4. Evolución del modelo [Lai et al., 2004] con $Shh/k_{Shh} = 0,1$ | 25 |
| 2.5. Evolución del modelo [Lai et al., 2004] con $Shh/k_{Shh} = 1,5$ | 26 |
| 2.6. Representación gráfica de los puntos de corte entre la recta Gli=Gli y la definida por (2.12). | 27 |

| | | |
|------|--|----|
| 2.7. | Representación gráfica del procedimiento para buscar ceros implementado. La recta representa la resta entre los miembros de (2.12). | 28 |
| 2.8. | Diagrama de Bifurcación de [Lai et al., 2004] con Gli frente a Shh/K_{Shh} . | 29 |
| 2.9. | Diagrama de Bifurcación de [Lai et al., 2004] con Gli frente a r_{g3b} | 30 |
| 3.1. | Variación del nuevo operador BEWARE | 43 |
| 3.2. | Variación del nuevo operador BEWARE en más rango | 44 |
| 3.3. | Evolución temporal del nuevo operador BEWARE | 45 |
| 3.4. | Representación gráfica de las rectas involucradas en la ecuación de punto fijo para el nuevo BEWARE | 45 |
| 3.5. | Representación gráfica del algoritmo de localización de ceros, que busca en qué puntos corta la recta generada por la resta entre los miembros de la ecuación de punto fijo generada con el nuevo BEWARE al eje de abcisas . | 46 |

Capítulo 1

Introducción

1.1. Motivación y objetivos

Durante el desarrollo humano, las células están expuestas a una compleja red de señales reguladoras las cuales deben interpretar correctamente para desarrollar las funcionalidades necesarias requeridas por el organismo. Por tanto, se pueden entender la transducción de señales y las cascadas de regulación genética como mecanismos de procesamiento de la información que traducen la información extracelular en decisiones intracelulares.

El presente trabajo pretende mostrar las diferencias en cuanto a comportamiento cualitativo que se pueden encontrar modelando estos complejos sistemas de regulación biológicos mediante distintas aproximaciones teóricas.

En particular, nos centraremos en el estudio del sistema de señalización del Sonic Hedgehog (en adelante Shh). El Shh es una proteína que conforma uno de los factores de señalización canónicos, secretados para regular la función celular y, por tanto, el desarrollo en numerosos sistemas.

Por ejemplo, la importancia del Shh se pone de manifiesto teniendo en cuenta algunos de sus muchos roles durante el desarrollo:

- Modela la diferenciación del tejido de la médula espinal.
- Modela la diferenciación del tejido de los brotes de .



Figura 1.1: Representación esquemática de la proteína de Shh. Fuente: [Wikipedia contributors, 2018b]

- Controla la diferenciación neuronal del mesencéfalo.
- Controla la diferenciación neuronal del prosencéfalo ventral.

Una de las características más importantes es que el Shh puede modelar distintos tejidos durante el desarrollo formando un gradiente de concentración [Saha and Schaffer, 2006]. Debido a este gradiente las células detectan su posición dentro del mismo y se diferencian en distintos fenotipos ¹ según la concentración y el tiempo de exposición a la concentración.

Aparte, como se destaca en [Lai et al., 2004] el Shh también controla la proliferación de numerosas poblaciones de células durante el desarrollo, incluidas las células granulares del cerebelo. Esto implica que las mutaciones dentro del sistema de señalización/regulación del Shh se han asociado con la proliferación de tumores (cáncer) en numerosos tejidos, como en el reciente artículo [Clement et al., 2007].

Con esta breve introducción ponemos de manifiesto la importancia de conocer el comportamiento de estas redes de señalización. Nuestro interés principal será conocer cómo afecta de manera cualitativa, un cambio en el procedimiento teórico de modelado de los mecanismo bioquímicos involucrados en la expresión genéticas. Centrándonos en redes de regulación/expresión que relacionan las proteínas *Ptc*, *Gli* y *Shh*.

¹Denominamos fenotipo a la expresión del genotipo, es decir, la expresión de los genes, en función de un determinado ambiente.

Acotando aún más el sujeto de estudio, los factores de transcripción dentro de la familia *Gli* desempeñan papeles críticos en la mediación e interpretación de las señales de Shh [Ruiz i Altaba, 1999]. Elucidar cómo funcionan nuestras redes de regulación y las proteínas *Gli* nos permitirá ampliar nuestro conocimiento de cómo las células proliferan, diferencian o sobreviven en respuesta a señales de *Shh*, procesos con importancia capital en una gran cantidad de aspectos como por ejemplo [Dahmane et al., 1997]. En especial es importante conocer de qué manera afectan estos cambios a la aparición/desaparición y/o existencia/inexistencia de estados estables o metaestables, y, por supuesto, de cómo están relacionados y cómo podemos llegar de unos a otros.

Nuestro trabajo recoge un estudio completo del modelo clásico propuesto en [Lai et al., 2004], aportando nueva información dentro del mismo, y un conjunto de experimentos numéricos relacionando nuevos desarrollos teóricos con el modelo clásico.

Además, al contrario que los artículos originales, todos los códigos se encuentran online y libres para su uso y reproducibilidad, vía archivos y vía *Jupyter Notebooks*.

Por otra parte, presentamos un estudio teórico y numérico de una nueva forma de modelar desde el enfoque termodinámico este proceso, propuesta en [Cambon and Sanchez,] para comparar las diferencias cualitativas entre ambos, y avanzar qué posibles resultados podríamos obtener de este nuevo modelo.

1.2. Sistema de señalización de Shh

En esta sección pretendemos ofrecer una visión general de la red de regulación de Shh que se observa en la célula. Todos los modelos usados dentro de este trabajo poseen puntos de vista compartidos, por lo que todas aquellas características que comparten ambos modelos se pueden encontrar aquí.

Así pues, se puede encontrar en esta sección la descripción bioquímica del sistema de señalización de Shh y las ecuaciones estándar empleadas en los procesos e interacciones bioquímicas que poseen ambos modelos.

1.2.1. Descripción bioquímica del proceso

La red de señalización de *Shh* comprende la actividad de varias proteínas 1.2 y genes :

- **Sonic Hedgehog (Shh)**. Gen: *shh*.²
- **Smoothened (Smo)**: Proteína de la superficie celular.
- **Patched (Ptc)**: Receptor de la superficie celular. Gen: *ptc*.
- **Factores de transcripción *Gli***:
 - **Gli**: Engloba a *Gli1* y *Gli2*, puesto que sus funciones son similares. Genes: *gli1*, *gli2*.
 - **Gli3**: Gen: *gli3*.
 - **Gli3R**: Resultado de la proteólisis³ de *Gli3*.

Además, según la estrategia al modelar, tendremos distintas descripciones de la actividad global de la **ARN polimerasa**.

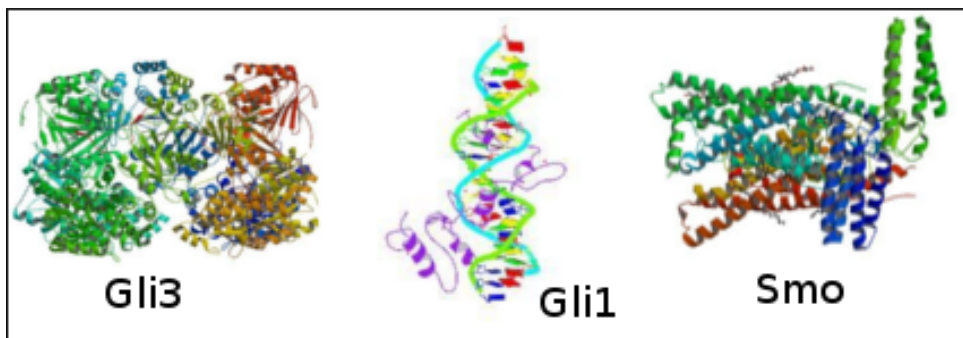


Figura 1.2: Representación esquemática de las proteínas Gli1, Gli3 y Smo. Fuente: [Hornbeck PV, 2015]

Presentamos el proceso de forma esquemática siguiendo las indicaciones de [Lai et al., 2004]:

1. El *Shh* interactúa con un receptor de superficie celular denominado *Patched (Ptc)*.
2. El *Ptc* inhibe la actividad de señalización de una segunda proteína de la superficie celular: *Smoothened (Smo)*.
3. La unión de *Shh* y *Ptc* neutraliza el efecto inhibitorio de *Ptc* sobre *Smo*.

²De forma convencional los genes que codifican una determinada proteína vienen expresados con el mismo nombre, pero en minúscula

³La proteólisis es la degradación de proteínas ya sea mediante enzimas específicas, llamadas peptidasas, o por medio de digestión intracelular.

4. Cuando *Smo* no está inhibido afecta a la actividad de la familia de factores de transcripción *Gli*.
5. En ausencia de *Shh*, *Gli3* es transformado mediante la proteólisis en *Gli3R* (represor de la transcripción génica).
6. Tras la señalización de *Shh* y *Smo*, la proteólisis se bloquea, lo que lleva a la acumulación de *Gli3*.
7. El *Gli3* (activador de la transcripción génica) activa la transcripción de los genes *gli1*, *gli2*, *ptc*, *shh*..
8. La activación de la transcripción de estos genes provoca la creación de *Gli* y *Ptc*, lo cual a su vez, favorece la generación de *Gli* y *Ptc*.

El valor añadido de incluir la ARN polimerasa en el modelo vendrá explicado en la sección 3 . En la figura 1.3 se puede encontrar un dibujo esquemático del proceso.

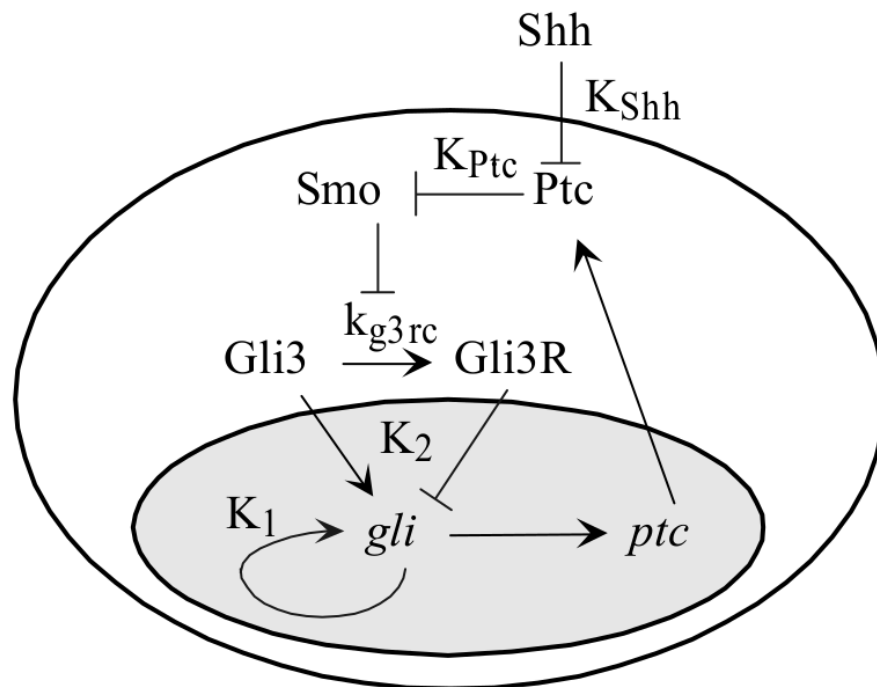
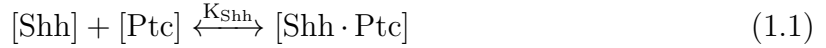


Figura 1.3: Representación esquemática de la red de transcripción. Fuente: [Lai et al., 2004]

1.2.2. Interacción entre Ptc y Shh

Shh y Ptc se unen de forma reversible con una constante de disociación K_- y de asociación K_+ . Al cociente entre ambas es lo que denominaremos K_{Shh} , mediante el siguiente esquema:



Además, asumimos que las uniones entre Ptc y Shh llegan rápidamente a un estado estacionario si tomamos la escala temporal de transcripción genética y síntesis de proteínas. Para conocer cual es, utilizamos la ecuación de Scatchard.

La ecuación de Scatchard es una ecuación utilizada en bioquímica y biología molecular para calcular la constante de afinidad de un ligando con una proteína, propuesta por primera vez en [Scatchard, 1949].

Sea una reacción como (1.1) tenemos, en virtud de la ley de acción de masas, que:

$$K_{Shh} = \frac{[Shh.Ptc]}{[Shh][Ptc]}$$

de donde,

$$[Shh.Ptc] = K_{Shh}[Shh][Ptc]$$

. Sea ahora ν representando los moles de ligando unido por mol de proteína, en primer lugar tenemos:

$$\nu = \frac{[Shh.Ptc]}{[Ptc_{Total}]} \quad (1.2)$$

Ahora bien, si operamos:

$$\nu = \frac{[Shh.Ptc]}{[Ptc_{Total}]} = \frac{[Shh.Ptc]}{[Ptc.Shh] + [Ptc]} = \frac{K_{Shh}[Shh][Ptc]}{[Ptc] + K_{Shh}[Shh][Ptc]} = \frac{K_{Shh}[Shh]}{1 + K_{Shh}[Shh]}$$

. Finalmente:

$$\nu = \frac{[Shh]}{[Shh] + K_{Shh}^{-1}}. \quad (1.3)$$

En este caso, uniendo (1.3) y (1.2) expresión:

$$[Shh.Ptc] = \frac{[Shh][Ptc_{Total}]}{K_{Shh}^{-1} + [Shh]}. \quad (1.4)$$

Si bien éste es el procedimiento empleado en [Lai et al., 2004], hacemos notar que la primera relación de Scatchard introducida se puede obtener basándonos en el estado de equilibrio de la ley de acción de masas, como avanzábamos. Añadimos, ya que se omiten en el artículo, las operaciones que nos llevan a este resultado intermedio.

Partiendo así de (1.1):

$$\frac{d}{dt}[Shh.Ptc] = k_+[Shh][Ptc] - K^-[Shh.Ptc]. \quad (1.5)$$

Si suponemos que el compuesto está en equilibrio:

$$\frac{d}{dt}[Shh.Ptc] = 0 = K_+[Shh][Ptc] - K^-[Shh.Ptc] \implies \frac{[Shh.Ptc]}{[Shh][Ptc]} = K_+/K_-. \quad (1.6)$$

Donde el cociente es denominado en las cuentas anteriores como: K_{Shh} . Además, para evitar complicaciones excesivas en la notación, denominaremos a K_{Shh}^{-1} como k_{Shh} .

1.2.3. Señal de transcripción

Vamos a considerar el término **Señal** como la fracción de Smo liberada de la inhibición del Ptc . Aunque Ptc y Smo no interactúan físicamente [Lai et al., 2004] propone modelarlo de manera similar a la unión de Shh y Ptc , puesto que la cantidad de Smo libre puede interpretarse como la cantidad que no está interactuando de forma eficiente con el Ptc . En este caso, tenemos:

$$[Shh.Ptc] = \frac{[Ptc_{libre}][Smo_{Total}]}{k_{Ptc} + [Ptc_{libre}]} \quad (1.7)$$

Donde Ptc_{libre} hace referencia al Ptc que no está interactuando con Shh y k_{Ptc} es la mitad de la concentración de Ptc necesaria para inhibir la actividad de Smo . Tal y como comentamos, definimos la **señal** (en adelante *Signal*) como:

$$Signal = \frac{[Smo_{libre}]}{[Smo_{Total}]} = \frac{[Smo_{total}] - [Smo.Ptc]}{[Smo_{Total}]} \quad (1.8)$$

Finalmente, usando (1.7) y (1.4) en (1.8) nos queda:

$$Signal = \frac{[Smolibre]}{[Smototal]} = \frac{\frac{Shh}{k_{shh}} + 1}{\frac{Shh}{k_{shh}} + 1 + \frac{Ptc}{k_{ptc}}} \quad (1.9)$$

1.2.4. Dinámica de Gli3 y Gli3R

Dinámica de Gli3

En ausencia de señalización Shh, Gli3 se escinde proteolíticamente en un fragmento que funciona como un represor transcripcional. En [Wang et al., 2000] muestran que el grado de proteólisis disminuye con el aumento de Shh. En este caso, imponemos que la tasa de proteólisis varíe inversamente con el nivel de señalización Shh en el sistema.

Así, nuestra la cantidad de Gli3 disminuye con una tasa k_{g3rc} que se modifica por la cantidad de *Signal* en el sistema y un parámetro de saturación K_{g3rc} .

A su vez, se ha demostrado que a medida que se activa la red de regulación génica, gli3 es transcripcionalmente reprimido [Wang et al., 2000]. Dos lecturas del grado de activación de nuestra red son Ptc y Gli.

Esto es importante, puesto que, aunque Ptc ofrece también una lectura del grado de activación, los resultados pueden variar en gran cantidad dependiendo de cual elijamos.

Por lo tanto, asumimos una relación inversa entre la transcripción de gli3 y la concentración de Gli en las ecuaciones para Gli3, partiendo de una tasa basal de generación de Gli3 que viene dada por la constante r_{g3b} .

Finalmente, con toda la información podemos entender como modelar matemáticamente la evolución de Gli3:

$$\frac{dGli_3}{dt} = \frac{r_{g3b}}{Gli} - k_{deg}Gli_3 - \left(\frac{k_{g3rc}}{K_{g3rc} + Signal} \right) Gli_3, \quad (1.10)$$

Dinámica de Gli3R

La existencia de esta molécula es completamente dependiente a la existencia de Gli3.

En su dinámica vamos a encontrar un término positivo exactamente igual a la rapidez en la que Gli3 es separado de forma proteolítica y, además, un término de degradación (cuya constante de degradación es igual a la constante de degradación de Gli3).

Esto nos deja con la expresión:

$$\frac{dGli3R}{dt} = \left(\frac{k_{g3rc}}{K_{g3rc} + Signal} \right) Gli3 - k_{deg} Gli3R, \quad (1.11)$$

1.3. Modelado BEWARE

Enfoques en el modelado de la regulación génica

El análisis detallado de las redes transcripcionales es clave para comprender los procesos biológicos centrales. Modelar correctamente la regulación de genes es fundamental para tal fin, puesto que la expresión génica está en el nexo de muchos procesos biológicos, y los cambios en los niveles de proteínas reguladoras o enlaces pueden ser la base de, por ejemplo, enfermedades de gran impacto como el cáncer.

A la hora de profundizar y aportar nuevo conocimiento en este área, las matemáticas se han desarrollado por diversos caminos, resaltando unas u otras características. Como se destaca en [Ay and Arnosti, 2011] dentro de este actual abanico de técnicas tenemos dos grandes estrategias iniciales: *Enfoque analítico o estadístico*.

Durante nuestro trabajo nos hemos centrado en el primero. Dentro del cual podemos encontrar tres grandes ramas: *modelos termodinámicos, booleanos y de ecuaciones diferenciales*. Cada una de las cuales debe ser tomada con cautela para obtener el máximo beneficio en cuanto al conocimiento del comportamiento cualitativo y cuantitativo de las soluciones.

Lo más habitual presentado en el grado y en el máster son los modelos basados en ecuaciones diferenciales, ya sean ordinarias o en derivadas parciales. Estos modelos surgen de la necesidad de crear sistemas dinámicos con muchas componentes que evolucionen a lo largo del tiempo. Como hemos visto en la sección anterior, esta técnica ha sido empleada por los dos modelos estudiados, en aquellos comportamientos dependientes de proteínas que modelaban la asociación y disociación de compuestos.

Sin embargo, a la hora de modelar el proceso de transcripción genética, tanto para *Gli* como para *Ptc*, el modelado va a seguir **el enfoque termodinámico**.

Antes de continuar vamos a ofrecer una breve descripción generalista del proceso general que planteamos modelar: la transcripción génica.

Biología de la transcripción génica

Según se expone en [Gilbert, 2000], el proceso general de transcripción es un proceso biológico complejo. En primer lugar la enzima **ARN polimerasa (ARNp)**, que forma una nueva molécula de ARN a partir del código proporcionado por el ADN, debe unirse al ADN del gen. Se adjunta en un lugar anterior al código genético que queremos transcribir. Esta zona se conoce como **promotor**.

La ARNp necesita de ayuda para unirse a esta sección. Con el fin de controlar este proceso aparecen conjuntos de proteínas llamadas **factores de transcripción**.⁴

Vamos a detenernos en esta parte, para comprender mejor los factores de transcripción, puesto que juegan un papel importante en el modelado. Un factor de transcripción típico se une al ADN en una determinada secuencia diana ⁵. Una vez que está unido, el factor de transcripción hace que sea más difícil o más fácil que la ARN polimerasa se una al promotor del gen. Con esto en mente, podemos clasificar este tipo de factores de activación, según faciliten o dificulten esta unión.

Por un lado algunos factores de transcripción activan la transcripción. Por ejemplo, pueden ayudar a que los factores de transcripción generales y / o la ARN polimerasa se unan al promotor.

Otros factores de transcripción reprimen la transcripción. Esta represión puede funcionar en una variedad de formas. Como un ejemplo, un represor puede interferir con los factores de transcripción basales o la ARN polimerasa, por lo que no pueden unirse al promotor o comenzar la transcripción.

Además, hay otro actor reseñable dentro de este procedimiento. Los **potenciadores** son

⁴Hay muchos tipos de factores de transcripción. Aunque su función en ocasiones puede ser desempeñada por factores de transcripción generales, existe una gran clase de factores de transcripción que controlan la expresión de genes individuales específicos.

⁵Los sitios de unión para los factores de transcripción a menudo están cerca del promotor de un gen. Sin embargo, se pueden encontrar en otras partes del ADN, a veces muy lejos del promotor, y aún afectan la transcripción del gen.

una secuencia de ADN que puede activar el uso de un promotor, controlando la eficiencia y la tasa de transcripción de ese promotor particular.

En general, estos son los factores a tener en cuenta en la transcripción génica. A modo de resumen dejamos la gráfica 1.4 y 1.5.

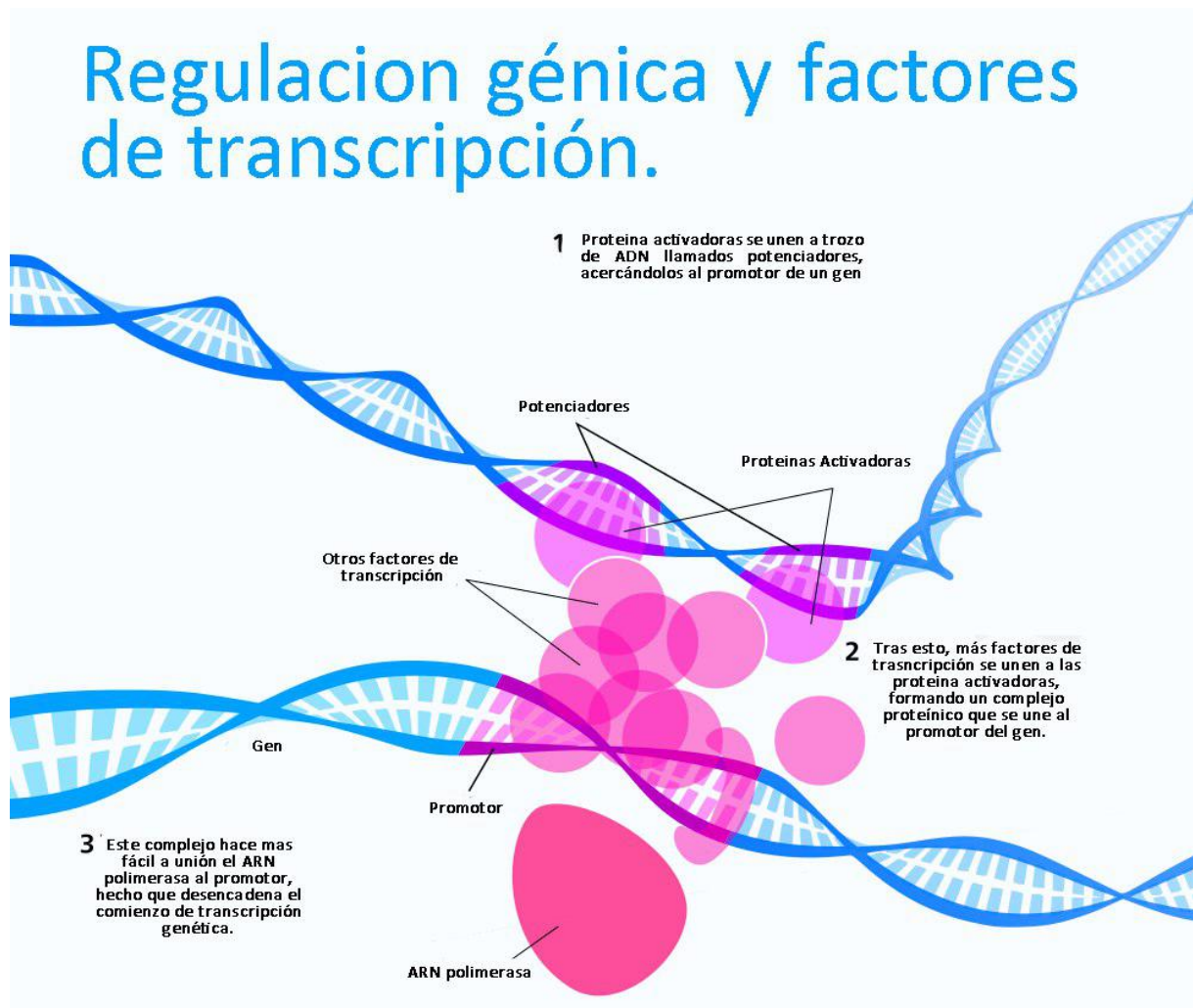


Figura 1.4: Representación artística de la transcripción génica, [Wikipedia contributors, 2018c]. Traducción y modificación propia

Modelado termodinámico

Este enfoque de modelado, como apunta [Ay and Arnosti, 2011], busca extraer información sobre la regulación génica a partir de las secuencias de las regiones reguladoras y la unión medida o inferida de los factores de transcripción específicos.

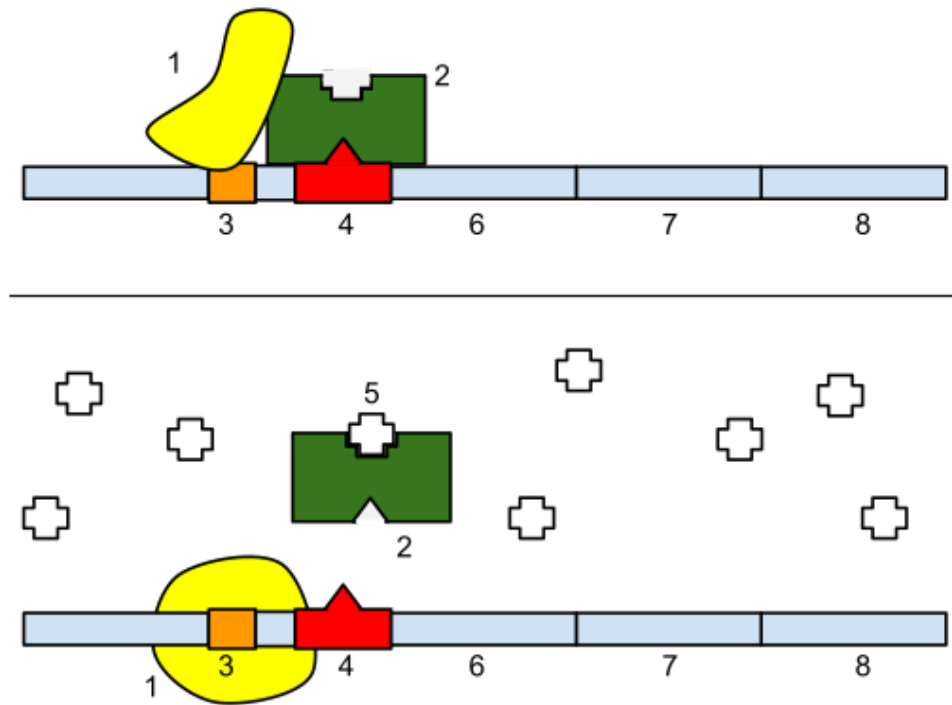


Figura 1.5: 1: ARN polimerasa, 2: represor, 3: promotor, 4: operador, 5: inhibidor del represor, 6-8: Genes. Arriba: el gen está esencialmente desactivado. Abajo: el gen está encendido [Wikipedia contributors, 2018a]. Traducción y modificación propia.

Es decir, supongamos que tenemos un promotor y algunos factores de transcripción que reprimen o promueven esta transcripción. En este caso, nuestro modelado quiere predecir cómo se activará o reprimirá la transcripción de un gen según las concentraciones de estos factores.

La clave fundamental modelando termodinámicamente es el cálculo de cómo las diferentes combinaciones de distintos lugares y números de unión en una región reguladora funcionan juntos para producir la expresión a lo largo del tiempo de un gen.

A grandes rasgos suponemos que la actividad del gen será mayor al aumentar al nivel de activadores unidos y disminuirá al aumentar el nivel de represores.

Procedimiento

Sea cual sea la estrategia a seguir, modelado termodinámico sigue dos pasos comunes a todos los modelos de esta rama que aun siendo comunes se pueden llevar a cabo de distintas maneras dando lugar a diversas versiones del modelado:

- En primer lugar: se enumeran todos los estados posibles del potenciador, en función de las posibles interacciones entre los factores de transcripción y el ADN, y se calcula un peso estadístico asignado a cada estado.
- El segundo lugar asignamos un nivel de expresión génica de cada estado. Los estados con alta ocupación de activadores son más proclives a inducir una expresión alta, mientras que la ocupación del represor puede dar como resultado una baja expresión.

Otro de los factores a tener en cuenta es el cálculo del peso estadístico para un estado. Para ello usamos la concentración de factores de transcripción y la afinidad de estos factores por sus sitios en el ADN. Para una unión abundante de proteínas a sitios de alta afinidad, el peso será mucho mayor que en los casos en que los factores de transcripción son escasos o el sitio de unión es débil.

La probabilidad de cada estado se puede calcular dividiendo el peso estadístico del estado por la suma del peso estadístico de todos los posibles estados.

Este proceso de cálculo puede incorporar propiedades que afectan la transcripción. Por ejemplo, interacciones cooperativas y competitivas entre factores de transcripción y los efectos inhibidores de los represores sobre los activadores se pueden agregar explícitamente al modelo asignando pesos más altos o más bajos.

Como podemos observar, estamos ante una forma de modelado que nos da bastante juego a la hora de modificar distintos parámetros y procedimientos. En particular, vamos a resaltar la mayor diferencia entre los distintos modelados que se han empleado en este trabajo:

- [Lai et al., 2004] modela la expresión génica como cantidad proporcional a la suma ponderada de los factores de transcripción (*enfoque stimulated*).
- Por otra parte, [Cambon and Sanchez, 2017] proponen que la expresión génica sea proporcional a la probabilidad de unión del ARN-polimerasa (*enfoque recruitment*), la cual viene modificada por los factores de transcripción.

Críticas recibidas

Finalmente, como último apunte, aunque partimos un de una forma de modelado con amplios resultados queremos resaltar algunas de las críticas que ha recibido esta forma de modelar.

La implementación actual ignora procesos adicionales como la estructura y modificación de la cromatina, o la metilación del ADN, y no trata de forma independiente el reclutamiento de cofactores o la maquinaria general de transcripción.

Se entiende actualmente que estos saltos teóricos no aportan gran información al sistema tal y como se expone en [Ay and Arnosti, 2011], aún así, creemos que es conveniente no perderlos de vista, debido a que es un cambio en los objetivos de los que se pretende modelar (nuestro nuevo modelo incluye la actividad de la ARN polimerasa) lo que ha supuesto un esperanzador cambio en el ajuste de este modelo con la realidad.

Capítulo 2

Modelo clásico

2.1. Introducción

El modelo que planteamos en esta sección pertenece a el modelado considerado clásico realizado en [Lai et al., 2004].

Como tenemos gran parte de la dinámica ya planteada, en esta sección, tal y como avanzamos al principio, vamos a presentar la forma en la que se modela la generación de Gli y Ptc (la transcripción genética) aplicando un enfoque de métodos de termodinámica estadística.

2.2. Modelado BEWARE

Partimos de dos resultados experimentales que muestran que *gli1*, *gli2* y *ptc* están regulados transcripcionalmente por la señalización de Shh.

Definimos K_1 como la constante de enlace de disociación de equilibrio de Gli y K_2 como la constante de enlace de disociación de equilibrio de *Gli*₃ (tanto activador como represor, Gli3R). Las zonas de unión al ADN de todas las formas de Gli están altamente relacionadas, lo que sugiere que estas las afinidades son similares.

Ante la decisión de qué cantidad de enlaces tomar, [Lai et al., 2004], para simplificar, suponen que hay el mismo número de posibles enlaces Gli dentro de los promotores para

Gli y Ptc.

El promotor puede existir en numerosos estados posibles (promotor vacío, dos Gli y un Gli_3 y el resto de combinaciones de 3 elementos). Además la probabilidad de cada estado de unión está determinada por las concentraciones relativas de las tres especies (Gli, Gli_3 , Gli_3R) y sus afinidades de unión al ADN.

Nuestro objetivo es desarrollar el modelo de acuerdo al procedimiento BEWARE, por tanto, para modelar el nivel de activación transcripcional del promotor, calculamos la suma de la probabilidad de cada posible estado del promotor multiplicado por una tasa de la activación de transcripción génica que la combinación particular induce. Sin embargo, para ello debemos determinar correctamente el nivel de activación para un estado dado, con este fin [Lai et al., 2004, Saha and Schaffer, 2006] aplican varias reglas:

- En primer lugar la unión del número máximo de activadores transcripcionales (una combinación de Gli y Gli_3) produce un estado con la máxima tasa transcripcional posible, igual a $(v_{max,G} + r_{bas})$ para el promotor gli.

En este caso v_{max} es la tasa de transcripción inducida máxima y r_{bas} es igual a la tasa basal de transcripción que se obtendría para un promotor completamente independiente. Implícita en esta expresión está la suposición de que [Lai et al., 2004] no tiene en cuenta la dinámica del ARNm, es decir, suponen que cada molécula de ARNm produce un número fijo de proteínas.

- A continuación, se permite la posibilidad de unión cooperativa de proteínas al promotor, de forma que el promotor con uno o más factores unidos tuviera una afinidad incrementada para el siguiente factor. A este factor lo denominamos *factor de cooperatividad de unión* = c que habitualmente viene igualado a la unidad ($c = 1$).

Además, para cada número de activadores unidos, menores que el número máximo de uniones posibles, la velocidad inducida $v_{max,G}$ se multiplica por un factor $e < 1$, para poner de manifiesto de una activación transcripcional menor que la máxima.

- Finalmente, para cada represor transcripcional Gli_3R unido, la suma $(v_{max,G} + r_{bas})$ se multiplica por el factor de represión $r < 1$.
- Con ambos elementos, multiplicamos la probabilidad de cada estado por la tasa de transcripción de cada estado, sumando los elementos resultantes entre sí y simplificando con ayuda del paquete de calculo simbólico desarrollado en [Meurer et al., 2017].

El resultado son dos expresiones relativas al proceso promotor y el proceso de transcripción basal:

Promoter =

$$\begin{aligned}
 & (GliK_2 + Gli_3K_1)(3K_1^2K_2^2e^2 + 3K_1K_2ce(GliK_2 + Gli_3K_1 + 2Gli3RK_1er) + c^2(Gli^2K_2^2 \\
 & + 3GliGli3RK_1K_2er + Gli_3^2K_1^2 + Gli_3K_1(2GliK_2 + 3Gli3RK_1er) + 3Gli3R^2K_1^2e^2r^2)) \\
 = & \frac{K_1^2K_2^2(3Gli_3K_1 + 3Gli3RK_1 + K_2(3Gli + K_1)) + 3K_1K_2c(GliK_2 + Gli_3K_1 + Gli3RK_1)^2 + c^2(GliK_2 + Gli_3K_1 + Gli3RK_1)^3}{(2.1)}
 \end{aligned}$$

Basal =

$$\begin{aligned}
 & K_1^2K_2^2(3GliK_2 + 3Gli_3K_1 + K_1(3Gli3Rr + K_2)) + 3K_1K_2c \\
 = & \frac{(GliK_2 + Gli_3K_1 + Gli3RK_1r)^2 + c^2(GliK_2 + Gli_3K_1 + Gli3RK_1r)^3}{K_1^2K_2^2(3Gli_3K_1 + 3Gli3RK_1 + K_2(3Gli + K_1)) + 3K_1K_2c(GliK_2 + Gli_3K_1 + Gli3RK_1)^2 + c^2(GliK_2 + Gli_3K_1 + Gli3RK_1)^3} \quad (2.2)
 \end{aligned}$$

2.3. Sistema final

Con los operadores BEWARE finalmente calculados podemos ya disponer de el sistema dinámico final que modeliza el sistema de señalización de Shh:

$$\frac{dGli}{dt} = v_{max,G}Promoter + r_{bas,G}Basal - k_{deg}Gli, \quad (2.3)$$

$$\frac{dGli_3}{dt} = \frac{r_{g3b}}{Ptc} - Gli_3 \left(k_{deg} + \frac{k_{g3rc}}{K_{g3rc} + Signal} \right), \quad (2.4)$$

$$\frac{dGli3R}{dt} = Gli_3 \left(\frac{k_{g3rc}}{K_{g3rc} + Signal} \right) - k_{deg}Gli3R, \quad (2.5)$$

$$\frac{dPtc}{dt} = v_{max,P}Promoter + r_{bas,P}Basal - k_{degp}Ptc. \quad (2.6)$$

2.4. Estados estacionarios

Siguiendo con el estudio estándar que se lleva a cabo en los modelos matemáticos procedemos con un estudio sobre los estados estacionarios que podemos encontrar en nuestro modelo. En primer lugar procedemos afrontando el problema desde una perspectiva analítica.

Sean las ecuaciones (2.3), (2.4), (2.5), (2.6), si suponemos que estas se encuentran en un estado estacionario entonces las concentraciones de las sustancias son constantes. Esto implica que su derivada temporal es igual a cero.

Dado que las ecuaciones contienen términos complejos, nos interesamos por agruparlas, de manera que los cálculos nos sean más sencillo en un primer intento de extraer información:

Por un lado de (2.3) y (2.6):

$$\begin{cases} 0 = v_{max,G}Promoter + r_{bas,G}Basal - k_{deg}Gli, \\ 0 = v_{max,P}Promoter + r_{bas,P}Basal - k_{degp}Ptc. \end{cases}$$

Teniendo en cuenta:

$$r_{bas,G} = \frac{v_{max,G}}{100}, r_{bas,P} = \frac{v_{max,P}}{100}.$$

Si igualamos ambas ecuaciones nos queda:

$$\frac{k_{deg}}{v_{max,G}}Gli = Promoter + \frac{1}{100}Basal = \frac{k_{degp}}{v_{max,P}}Ptc \implies \frac{v_{max,P}k_{deg}}{v_{max,G}k_{degp}}Gli = Ptc.$$

En particular si llamamos $k_{cc} = \frac{v_{max,P}k_{deg}}{v_{max,G}k_{degp}}$:

$$k_{cc}Gli = Ptc. \tag{2.7}$$

Por otra parte, de (2.4) y (2.5):

$$\begin{cases} 0 = \frac{r_{g3b}}{Gli} - Gli_3 \left(k_{deg} + \frac{k_{g3rc}}{K_{g3rc} + Signal} \right), \\ 0 = Gli_3 \left(\frac{k_{g3rc}}{K_{g3rc} + Signal} \right) - k_{deg}Gli_3R. \end{cases}$$

Sumando, obtenemos:

$$\begin{aligned} 0 = \frac{r_{g3b}}{Gli} - Gli_3k_{deg} - k_{deg}Gli3R &\implies \frac{r_{g3b}}{Gli} = Gli_3k_{deg} + k_{deg}Gli3R \implies \\ &\implies \frac{r_{g3b}}{Gli_3k_{deg} + k_{deg}Gli3R} = Gli. \end{aligned} \quad (2.8)$$

Con estas cuentas, podemos obtener, en primer lugar, una función de *Signal* (??) modificada gracias a (2.7), la llamaremos *Signal_{modificada}*:

$$Signal_{modificada} = \frac{\frac{Shh}{k_{shh}} + 1}{\frac{Shh}{k_{shh}} + 1 + \frac{k_{cc}}{k_{ptc}Gli}}. \quad (2.9)$$

Ahora, sustituimos los valores que tenemos de manera que podamos expresar todas las concentraciones en función de Gli.

Nuestro objetivo es intentar hallar los estados estacionarios mediante los puntos fijos entre dos expresiones de Gli. Con ello, usando (2.8) nos quedaría:

$$\frac{r_{g3b}}{Gli} = Gli_3k_{deg} + k_{deg}Gli3R \implies Gli_3 = \frac{r_{g3b}(K_{g3rc} + Signal_{modificada})}{k_{deg}(K_{g3rc} + Signal_{modificada})Gli}. \quad (2.10)$$

Y de nuevo, por (2.8):

$$Gli3R = \frac{r_{g3b}}{k_{deg}Gli} - Gli_3. \quad (2.11)$$

Debido a la capacidad de expresar Gli3R y Gli_3 con respecto a Gli, podemos obtener la variación de Promoter y Basal directamente con Gli, substituyendo en ellos el valor de Gli3R y Gli_3 . Con ello, finalmente obtenemos una igualdad cuyos puntos fijos nos darán los posibles estados estacionarios. Igualando 2.3 a cero obtenemos la siguiente ecuación de punto fijo :

$$Gli = \frac{v_{max,G}}{k_{deg}} Promoter_{modificado}(Gli) + \frac{1}{100} Basal_{modificado}(Gli). \quad (2.12)$$

2.5. Simulaciones

En esta sección vamos a desarrollar todas las simulaciones numéricas llevadas a cabo en el estudio cualitativo del modelo de [Lai et al., 2004]. Para estudiar este modelo y reproducir algunos resultados, seguimos el siguiente esquema:

- **Recolección y contraste de los parámetros** usados
- **Estudio y comparación de la variabilidad del operador BEWARE.** Dentro de este apartado comparamos numéricamente las reducciones desarrolladas en [Cam-bon and Sanchez,] con el fin de asegurar la exactitud de las mismas y, finalmente, usar la forma reducida para obtener programas más eficientes.
- **Análisis numérico de las soluciones estacionarias:** Desarrollamos la fórmula analítica para obtener un código que nos permita rastrear cambios en el comportamiento cualitativo ante grandes variaciones en los parámetros.
- En aquellos parámetros con especial interés por la reproducibilidad o la novedad en su estudio, **computamos un diagrama de bifurcación.**

2.5.1. Parámetros

A menos que se especifique lo contrario, para las simulaciones hemos tomado como valores de parámetros los expuestos en la tabla 2.1.

2.5.2. Variación del operador BEWARE (promoter y basal)

En primer lugar antes de comenzar la simulación completa comprobamos la reproducibilidad de los operadores que conforma el modelado BEWARE.

Además, pretendíamos tener una base para comparar posteriormente estos comportamientos con el nuevo operador.

Dentro las figuras se puede apreciar una línea discontinua. Esta línea marca el comportamiento asintótico del nuevo operador BEWARE a modo de adelanto de la siguiente sección. Para repetir las simulaciones, hemos computado la variación de Promoter 2.1 y Basal 2.2 con $Gli3 = 0$ y un cambio en la cantidad de Gli3R. Este método es el utilizado

| Tabla de parámetros de [Lai et al., 2004] | | | |
|---|------------------------------|--|----------------------------------|
| Parámetro | Valor | Descripción | Fuente |
| Shh | $0 - 30$ | Cantidad de Shh | [Lai et al., 2004] |
| k_{Shh} | $0,58 - 2,0nM$ | Constante de disociación de los enlaces Ptc-Shh | [Taipale et al., 2002] |
| k_{Ptc} | $8,3 \times 10^{-11}M$ | Mitad de la máxima concentración de Ptc que inhibe la señal de Smo | [Taipale et al., 2002] |
| k_{deg} | $0,009min^{-1}$ | Constante de degradación de todas las moléculas Gli | [Chen et al., 1999] |
| k_{g3rc} | $0,012min^{-1}$ | Constante de conversión de Gli_3 en Gli3R | [Lai et al., 2004] |
| r_{g3b} | $1,6 \times 10^{-19}M^2/min$ | Tasa basal de síntesis de Gli_3 | [Lai et al., 2004] |
| K_{g3rc} | $0,1$ | Constante de sensibilidad de la conversión a fuerza de la señal | [Lai et al., 2004] |
| k_{degp} | $0,09min^{-1}$ | constante de degradación de Ptc | [French and Lauffenburger, 1996] |

Cuadro 2.1: Tabla de parámetros de [Lai et al., 2004]

en [Lai et al., 2004] para visualizar el comportamiento de nuestros operadores y, para asegurar la buena reproducción del artículo, y la comparación con nuestros resultados, es el mismo utilizado para observar el comportamiento del nuevo operador BEWARE.

Reducción de la complejidad del operador

En los siguientes apartados tendremos que hacer grandes volúmenes de cuentas para estudiar variaciones en el comportamiento cualitativo del modelo. Por este motivo cualquier simplificación u optimización es altamente bienvenida a la hora de programar.

En nuestro caso, basándonos en los resultados de [Cambon and Sanchez,] tuvimos acceso a una simplificación de los operadores promotor y basal. Antes de realizar el estudio, sin embargo, quisimos comprobar numéricamente la viabilidad frente a múltiples variaciones de las dos expresiones.

Prueba de ello es la figura 2.3, donde mostramos como ambas expresiones son equivalentes cualesquiera variaciones hagamos. Dentro de esta tenemos en color la expresión clásica de los operadores y en líneas discontinuas mostramos la nueva expresión. Como se ve, la coincidencia es perfecta.

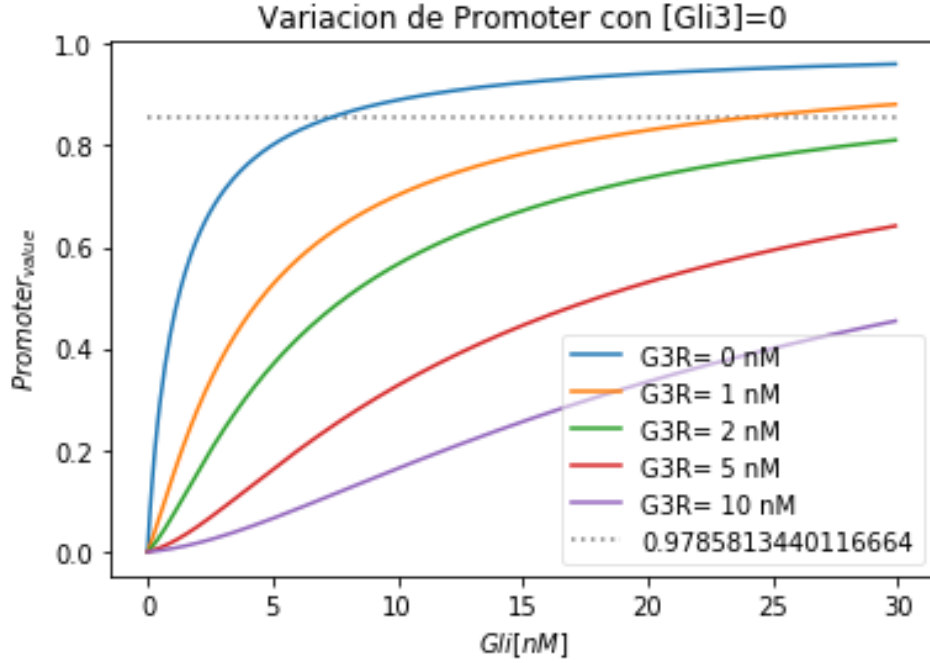


Figura 2.1: Variacion del Operador Promoter bajo la variacion de Gli3R

2.5.3. Evolución temporal

Una vez computamos los operadores en su forma reducida, procedimos a simular la evolución temporal del sistema.

Para ello utilizamos el módulo de cálculo numérico de Python y el algoritmo *lsode* que es robusto frente a comportamientos *stiff*. (Estos cálculos además han sido validados y reproducidos con Octave).

En particular corrimos simulaciones con distintas condiciones iniciales y distintos valores de Shh:

- Condiciones iniciales: $Gli=0.001$ $Gli_3=0$ $Gli3R=0$ $Ptc=0$. Valor Shh/K_{Shh} : 0,1
- Condiciones iniciales: $Gli=0.001$ $Gli_3=0$ $Gli3R=0$ $Ptc=0$. Valor Shh/K_{Shh} : 1,5
- Condiciones iniciales: $Gli=0.001$ $Gli_3=0$ $Gli3R=0$ $Ptc=0$. Valor Shh/K_{Shh} : 15
- Condiciones iniciales: $Gli=1$ $Gli_3=1$ $Gli3R=1$ $Ptc=1$. Valor Shh/K_{Shh} : 0,1
- Condiciones iniciales: $Gli=0.001$ $Gli_3=10$ $Gli3R=10$ $Ptc=10$. Valor Shh/K_{Shh} : 0,1

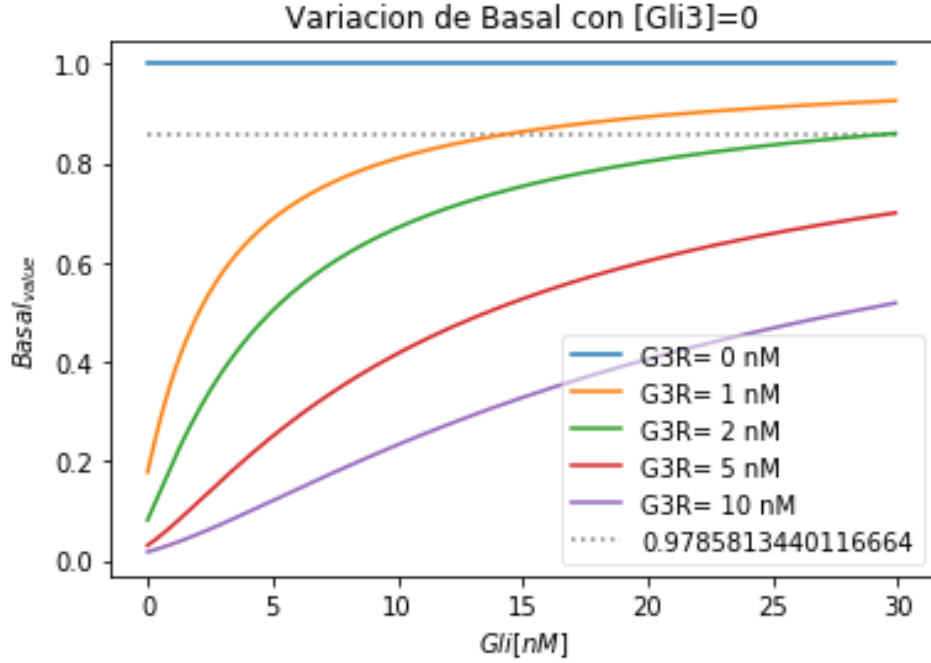


Figura 2.2: Variacion del Operador Basal bajo la variación de Gli3R

Dentro de estas simulaciones preliminares, en cuanto al estudio cualitativo, observamos dos interesantes comportamientos según varía la cantidad de $Shh/k_{Shh} = 0,1$ en 2.4, $Shh/k_{Shh} = 1,5$ en 2.5. Esto nos confirma la multiestabilidad del sistema que se afirma en el artículo original. Pero, como veremos más adelante, esta debe ser tomada con precaución.

En la Figura 2.4, podemos observar que si las cantidades de Shh disminuyen en gran medida, tanto Gli como Ptc se van a ver afectados, disminuyendo su cantidad. Esto se debe al aumento de Gli3R en nuestro sistema puesto que el principal inhibidor de la proteólisis de Gli_3 en Gli3R es Shh y este se encuentra en baja concentración.

Por contraparte, en la figura 2.5, la proporción Shh/K_{Shh} aumenta en gran medida. Esto activa toda la maquinaria bioquímica del proceso, inhibiendo Gli3R, que cae en mínimos y produciendo grandes cantidades de Gli y Ptc.

Los estados estacionarios sirvieron también de punto de partida para nuevas simulaciones, sin embargo, hay resultados de [Lai et al., 2004] que hemos concluido irreproducibles, destacando la dinámica del sistema con un cambio brusco de la cantidad de Shh¹. Por su parte, la dinámica del sistema tampoco se encuentra estudiada, y se presentan en el

¹Figura 2(A) de [Lai et al., 2004].

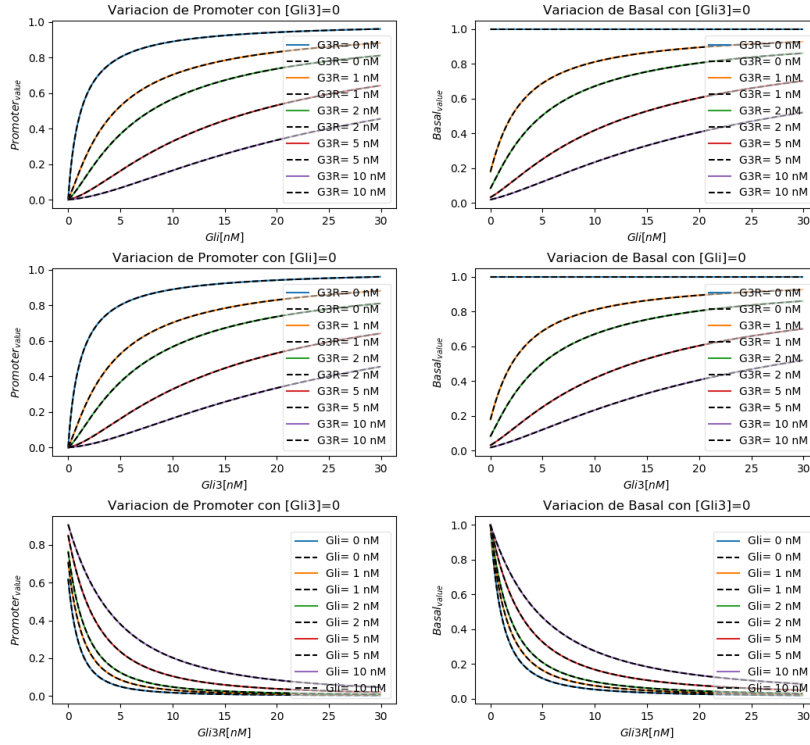


Figura 2.3: Comparación numérica de las formas de reducidas de Promoter y Basal (en negro discontinuo) frente a las formas de [Lai et al., 2004]

artículo solo comportamientos de Gli.

2.5.4. Análisis numérico de los estados estacionarios

Para el análisis numérico de los estados estacionarios seguimos dos estrategias. En primer lugar programamos las expresiones analíticas obtenidas en el apartado 2.4. Con ellas podemos representar la función y la recta $y = Gli$ (tal y como se muestra en la figura 2.6).

Esta representación nos permite observar los puntos de corte de ambas funciones y, con ello inferir los estados estacionarios. Más aun, nos permite observar el comportamiento de la curva más claramente según varían los parámetros para decidir como alterarlos de manera que se produzcan puntos de corte (lo que nos dio una primera intuición sobre r_{g3b}).

Con esta primera parte, para analizar grandes volúmenes de variaciones de parámetros cambiamos el enfoque. Dado que calculamos de forma discreta, escribimos un programa

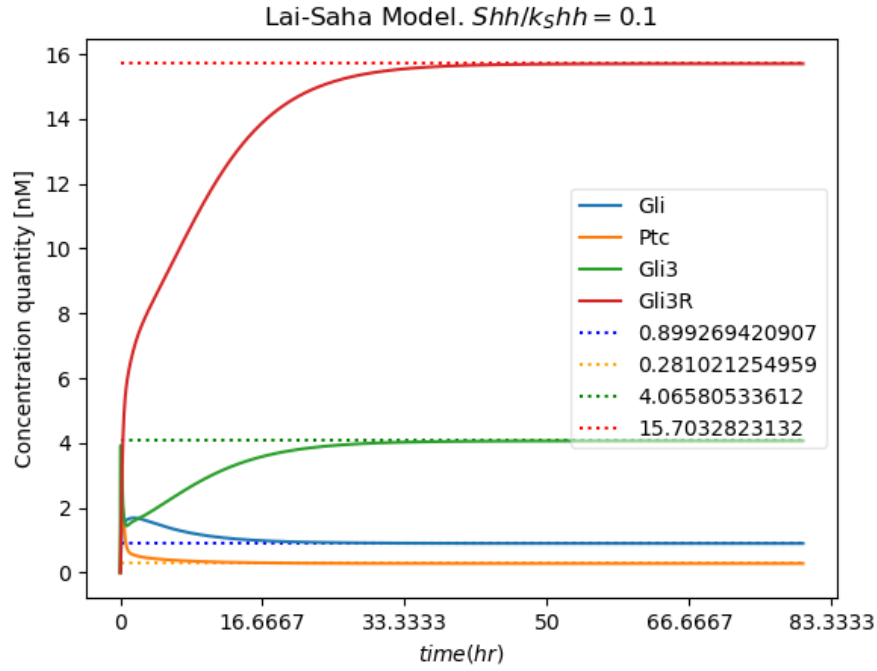


Figura 2.4: Evolución del modelo [Lai et al., 2004] con $Shh/k_{Shh} = 0,1$

que computaba la resta de ambas funciones y nos devuelve el número de veces que la resta cambia de signo (es decir, el punto i -ésimo es mayor que cero y el i -ésimo +1 es menor, o viceversa). El programa produce un *log*, donde se pueden consultar todos los resultados, y guarda en un fichero a parte aquellos que tengan un número de ceros mayor del configurado. Extracto del *log* puede verse así:

```
Shh = 0.1 r_g3c=1 Zeros:1
```

En la Figura 2.7, se puede observar una representación gráfica de los puntos que buscaría nuestro algoritmo. Con todo ello, obtuvimos intuición sobre que diagramas de bifurcación computar, y obtuvimos el código base para explorar el nuevo modelo.

2.5.5. Diagramas de bifurcación

Durante el estudio de este modelo procedimos también a realizar diagramas de bifurcación² con aquellos parámetros que el análisis numérico nos mostró como relevantes. Además,

²Un diagrama de bifurcación de un sistema dinámico es una estratificación de su espacio de parámetros inducida por la equivalencia topológica, junto con los retratos de fase representativos de cada estrato.

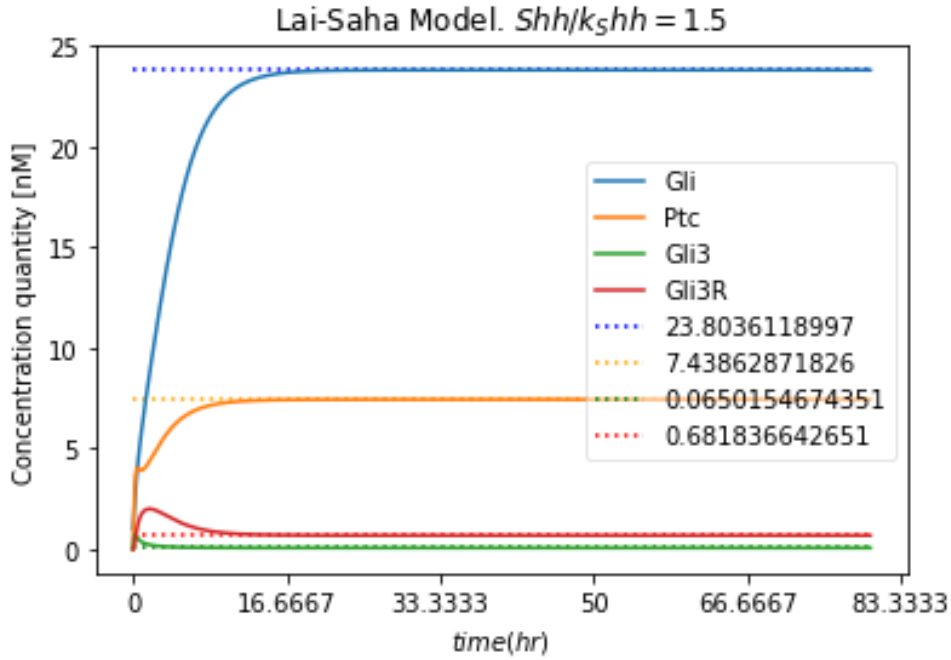


Figura 2.5: Evolución del modelo [Lai et al., 2004] con $Shh/k_{Shh} = 1,5$

procedimos a evaluar la reproducibilidad del artículo en los diagramas que presentan.

La herramienta escogida para tal fin fue AUTO, un motor para el cálculo de diagrama de bifurcaciones. En particular utilizamos XppAut, un intérprete de AUTO que nos permite realizar continuación, cambios de ramas y diagramas de bifurcación completos, así como el cálculo de la estabilidad de los puntos fijos.

Para una mayor profundización en el análisis numérico de bifurcaciones puede consultarse [Meijer et al., 2012]. Además, en mi trabajo de fin de grado [Ortiz, 2017] puede encontrarse una guía básica de AUTO para interpretar los resultados que imprime por pantalla (se maneja desde la terminal) y como pasarlos a un diagrama. Señalamos que la generación de diagramas de bifurcación es un tema denso y de alta sensibilidad numérica, por lo que recomendamos visitar los dos documentos citados.

Exponemos también, brevemente la técnica empleada para la obtención:

1. Comenzamos utilizando estados cercanos a los estados estacionarios como punto de partida para la iteración temporal³.

³Es importante señalar que la interacción temporal usa un jacobiano calculado por la máquina, no introducido manualmente (pues AUTO ofrece esta posibilidad)

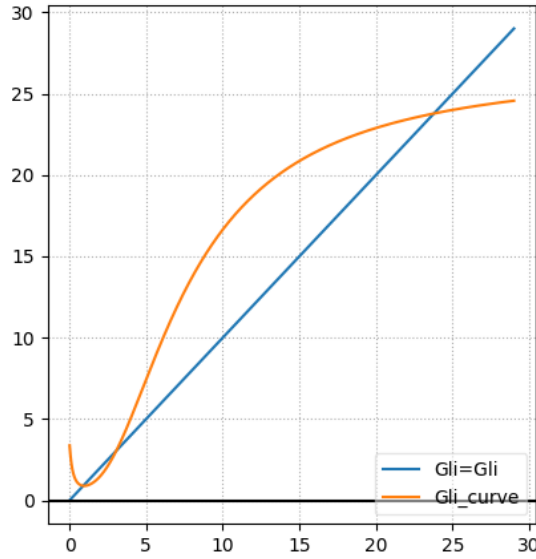


Figura 2.6: Representación gráfica de los puntos de corte entre la recta $G_{li}=G_{li}$ y la definida por (2.12).

2. Posteriormente a la iteración en tiempo, alcanzamos un valor estacionario con una sensibilidad previamente introducida a la máquina.
3. Utilizamos XppAut para obtener la estabilidad de la solución obtenida. En caso de ser estable, procedemos a exportar el resultado a AUTO, donde generaremos el diagrama.
4. En AUTO ajustamos los parámetros numéricos del programa. En particular el tamaño de paso (positivo para generar ramas según aumenta el parámetro y negativo para hacerlo según disminuye).
5. Podemos adecuar también el salto entre las ramas del diagrama para englobar posibles comportamientos extraños.
6. Finalmente generamos un archivo AUTO con la información del diagrama y, inmediatamente después, el diagrama.

Bifurcación bajo Shh/K_{Shh}

Debido a las diferencias que pudimos encontrar durante el desarrollo del modelo, nos pareció interesante intentar reproducir el comportamiento de G_{li} frente a la variación de

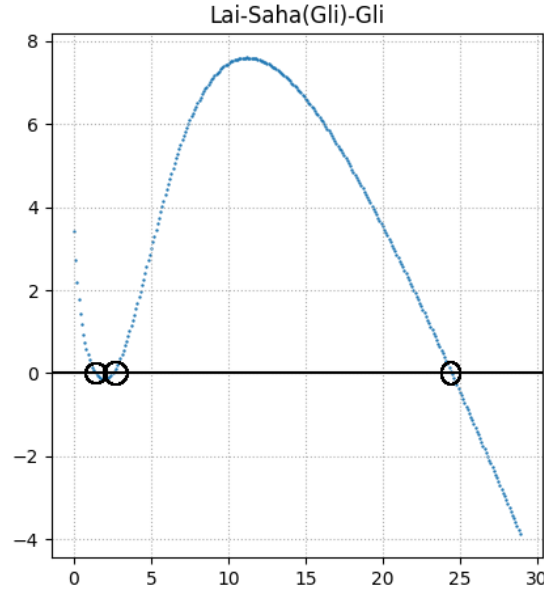


Figura 2.7: Representación gráfica del procedimiento para buscar ceros implementado. La recta representa la resta entre los miembros de (2.12).

Shh/K_{Shh} . En la figura 2.8 se puede encontrar el diagrama hallado. En ese caso las líneas rojas representan estados estables y las negras inestables.

En este caso, frente a los resultados del paper, obtenemos un comportamiento de interruptor biestable, sin embargo este no es reversible si no irreversible⁴. Es este mismo hecho el que motiva que las líneas del diagrama sobrepasen el 0 para valores de Shh cuando estamos retrocediendo.

Estos resultados, aunque muestran una dinámica similar, difieren cualitativamente de [Lai et al., 2004]. Sin embargo una revisión bibliográfica nos mostró como se relacionan con los posteriores trabajos del equipo, en concreto [Saha and Schaffer, 2006].

Tendremos, a vista del diagrama, una fuerte rama estable en creciente hacia los 25nmol de Gli y, podemos encontrar otro estado estable con una menor cantidad de Gli según crece la cantidad de Shh hacia 1,8. Es de esperar que Gli acabe cayendo en la región estable superior.

⁴Esta denominación es la elegida por los autores del artículo para denominar aquel comportamiento biestable que no puede cambiar entre estados.

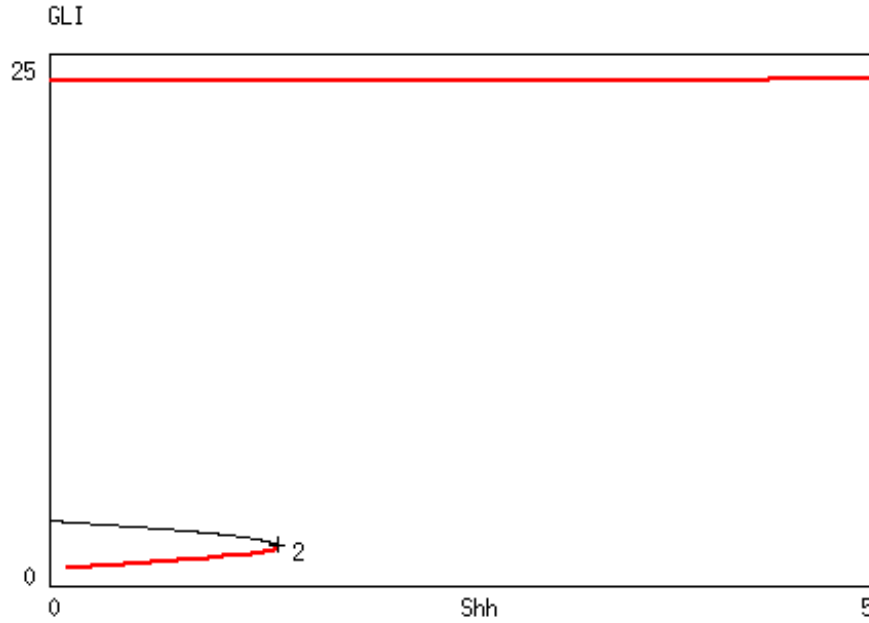


Figura 2.8: Diagrama de Bifurcación de [Lai et al., 2004] con Gli frente a Shh/K_{Shh}

Bifurcación bajo r_{g3b}

Durante el desarrollo de los estados estacionarios, pudimos observar como las variaciones en la tasa de sintetización basal del Gli_3 afectaban dramáticamente a los puntos de corte.

Con ello, teniendo como referencia los códigos del apéndice A, desarrollamos el diagrama de bifurcaciones de Gli frente a la variación de este parámetro en la figura 2.9, algo que no tenemos constancia de que se hubiese llevado a cabo.

En general, concluimos que la tasa de generación de Gli_3 juega un papel fundamental en la dinámica de este modelo, tanto por su posible alteración escogiendo como elemento delimitante Gli o Ptc , como por la variación de este en sí misma.

2.6. Críticas

Aunque el modelo es un referente desde hace muchos años, hemos encontrado dificultades a la hora de estudiarlo. Exponemos una breve crítica al mismo de carácter constructivo.

- Posibles erratas: Si bien el comportamiento es el esperado y descrito en la mayoría de los casos, ha sido habitual durante la investigación encontrarnos con problemas a

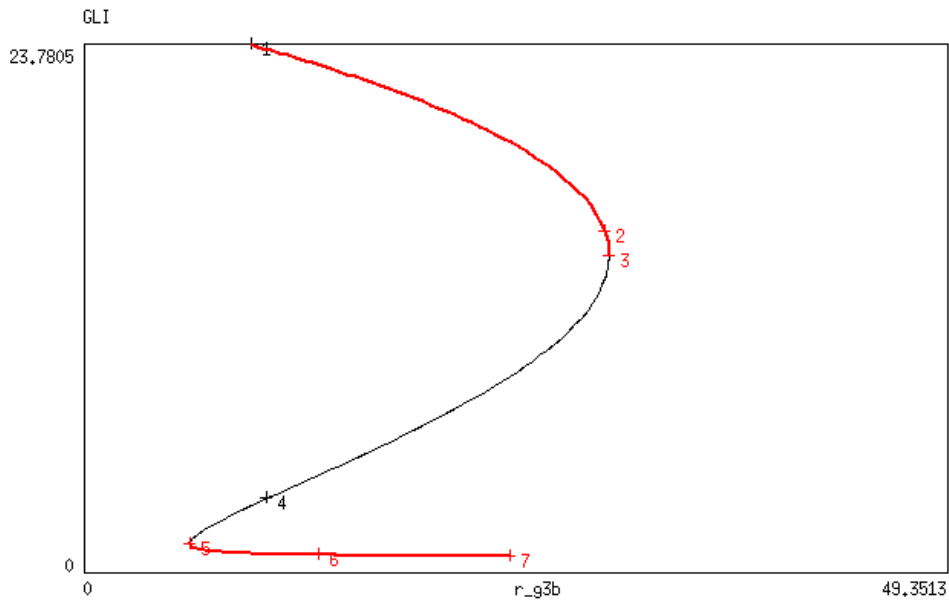


Figura 2.9: Diagrama de Bifurcación de [Lai et al., 2004] con Gli frente a r_{g3b}

la hora de reproducir los resultados que vienen descritos en el artículo. La ausencia de datos abiertos para esta tarea, la ausencia del código usado en muchos casos y, de código privativo en otros, dificulta en gran medida el estudio y reproducción. Dejamos patente que hay gráficas (corregidas en otros artículos) y expresiones con algunas erratas que pueden llevar a error.

- Algunas simplificaciones del modelo necesarias para el desarrollo del mismo pueden ser atacadas para estudiarlo en más profundidad. En particular, es nuestro enfoque añadir detalles que este modelo no contempla con el fin de estudiar si así podemos comprender mejor este proceso.

Capítulo 3

Modelo alternativo

3.1. Introducción

Como adelantábamos en la sección 1.3 si nos centramos en el modelado BEWARE de la transcripción génica esencialmente tenemos dos enfoques: modelar la expresión génica como una cantidad proporcional a la suma ponderada de los factores de transcripción o hacerlo proporcional a la probabilidad de unión del ARN polimerasa, que va modificada por los factores de transcripción.

Tras extenso estudio que se ha hecho del modelo [Lai et al., 2004], la multiestabilidad está más que demostrada para conjuntos de valores de parámetros esperables, así como para un conjunto de condiciones iniciales que también podrían esperarse en situaciones biológicas.

Aun así, la multiestabilidad que se pone de manifiesto en el estudio del modelo presenta dificultades a la hora de ser contrastada con experimentos biológicos. Creemos por tanto que un nuevo enfoque puede ser beneficioso para captar las sutilezas de la realidad biológica.

Es decir, en experimentos biológicos no observamos este tipo de multiestabilidad. Por tanto si queremos que nuestro modelo de transcripción represente fielmente el comportamiento de la transcripción genética debemos modificarlo o cambiarlo.

Así, antes de cambiar grandes características de nuestro modelo, nace la idea de modificar nuestra forma de crear el operador BEWARE usando el segundo enfoque que tiene en

cuenta el ARN polimerasa. Para ello utilizaremos el enfoque de [Cambon and Sanchez, 2017].

Tras el esquema de modelado, vamos a presentar los experimentos numéricos que nos llevan a pensar que podemos estar ante un comportamiento más fiel a lo observado en biología con sólo un estado estable.

3.2. Modelado BEWARE

Comenzamos aplicando las ideas del método termodinámico a nuestros dos genes, *gli* y *ptc*, los cuales suponemos controlados por tres factores de transcripción $\{Gli, Gli_3, Gli3R\}$ que son activador, activador y represor, respectivamente.

Consideraciones iniciales

- Destacamos de nuevo, antes de comenzar el resto del cálculo del operador, que la expresión de la evolución de la cantidad de las proteínas generadas por *gli* y *ptc*, *Gli* y *Ptc*, vendrá disminuida por la degradación natural de estas moléculas con dos constantes de degradación. Esto implica que nuestro resultado final será un modelo de la forma:

$$\frac{dGli}{dt} = BEWARE([Gli], [Gli_3], [Gli3R]) - k_{deg}Gli \quad (3.1)$$

e igual con el *Ptc*.

- En el modelo, además, las reacciones de unión de los factores de transcripción y del ARN polimerasa son mucho más rápidos que la síntesis de la proteína *Gli* o *Ptc*, por lo tanto, consideramos en equilibrio termodinámico dado por la Ley de Acción de Masas.
- Por otra parte en este trabajo se considera una versión del operador con cooperatividad total/ausencia de cooperatividad entre factores de transcripción. Podríamos estudiar cómo afecta la hipótesis de tener una cooperatividad parcial en futuros desarrollos del mismo, siguiendo las indicaciones de [Cambon and Sanchez, 2017].

3.2.1. Cálculo del operador

Nuestro objetivo es averiguar la expresión del operador BEWARE en 3.1. Partiendo de la segunda consideración inicial, calculamos la expresión de los primeros complejos que se formarían al unir a una región de regulación vacía ($[B]$ de ahora en adelante) con alguno de los promotores (Gli y Gli_3), represores ($Gli3R$) o ARN polimerasa. En ese caso, tienen una concentración en el equilibrio termodinámico expresada como:

$$[BW] = \frac{K_{+W}^{(1)}}{K_{-W}^{(1)}} [W][B] := \frac{[W]}{K_W^{(1)}} [B] \quad (3.2)$$

donde W representa cualquier activador, represor o ARN polimerasa y $K_W^{(1)}$ es la constante de disociación. Además, siguiendo la notación de [Cambon and Sanchez, 2017], el superíndice (1) representa que no hay otro factor de transcripción en el momento de la unión en el complejo.

Nuestro enfoque principal ocurre en una situación de cooperatividad¹. Por tanto, cuando nuevos factores se unan al complejo su constante de disociación vendrán modificada por los anteriores. Esto es, sea la reacción:



las concentraciones de equilibrio tendrían dadas siguiendo el equilibrio termodinámico:

$$[BWW] = \frac{[W][W]}{K_W^{(2)} K_W^{(1)}} [B]. \quad (3.4)$$

Sin embargo, la cooperatividad se refleja si imponemos:

$$K_i^{(2)} = K_k^{(1)} / c \forall i, k = Gli, Gli_3, Gli3R, ARNP,$$

donde c es una constante positiva mayor que 1.

En particular, nuestro modelado está enmarcado en la situación de cooperatividad total,

¹Si se produce la cooperación, sería necesario saber qué factores de transcripción se ven afectados por otros puesto que el la concentración del equilibrio dependerá de estas relaciones. En nuestro caso particular, sí podemos usar este enfoque.

esto implica que cuando se une un factor de transcripción a nuestro complejo este modifica las afinidades de todos los siguientes de la misma manera.

Notemos n el número de posiciones de unión (en nuestro caso $n=3$) de manera que j_A sea el número de activadores unidos ($j_{gGi} + j_{Gli3}$), $j_r = j_{Gli3R}$ el número de represores y $j_0 = n - j_A - j_R$ como el número de espacios libres.

Con ello, para calcular el estado de equilibrio de cualquier disposición, tenemos:

$$[BGli^{j_{Gli}} Gli_3^{j_{Gli3}} Gli3R^{j_{Gli3R}}] = [B]c^{(j_A+j_R-1)+} \left(\frac{[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{[Gli_3]}{K_{Gli3}} \right)^{j_{gli3}} \left(\frac{[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}}. \quad (3.5)$$

Donde $+$ denota la función parte positiva, puesto que la cooperatividad no tiene lugar si no hay 2 o mas factores de transcripción cooperativos en la configuración.

Con respecto al proceso de unión de ARN polimerasa, los factores de transcripción trabajan juntos tratando de activar o reprimir el proceso de unión por un mecanismo conocido como reclutamiento. Por lo tanto, consideramos que los activadores interactúan con RNAP con una interacción “adhesiva” que da lugar a una modificación de la afinidad de la unión de la ARN polimerasa: K_{RP}/a^{j_A} donde a es una constante de cooperatividad mayor que 1. Por el contrario, el efecto de los represores se modela en términos de una interacción “repulsiva” que modifica el afinidad de unión K_{RP}/r^{j_R} con un factor anti-cooperatividad r menor que 1.

Espacio de todas las configuraciones

Para avanzar en la obtención del operador BEWARE siguiendo el esquema, comenzamos calculando el espacio de todas las posibles configuraciones. Sea un complejo compuesto por $j_{Gli}, j_{Gli3}, j_{Gli3R}, j_0, j_P$ activadores, represores, espacios vacíos y ARN polimerasa. En particular j_p puede ser 0 o 1 según encontremos o no ARNp en el compuesto.

Supongamos que no hay ningún ARN polimerasa en nuestro complejo (esto es, $j_p = 0$). En ese caso el estado de equilibrio vendría dado por la expresión:

$$C(C)[B] \left(\frac{[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{[Gli_3]}{K_{Gli3}} \right)^{j_{gli3}} \left(\frac{[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}}, \quad (3.6)$$

donde $C(C)$ hace referencia a $c^{(j_A+j_R-1)+}$. Además, como hemos dicho, nuestro modelo no tiene en cuenta la disposición espacial, con lo que este estado puede venir dado por todas

las combinaciones espaciales del número de activadores, represores y espacios vacíos que nos indica $j_{Gli}, j_{Gli_3}, j_{Gli_3R}, j_0$. Por tanto, multiplicamos este estado por todas la combinaciones que pueden darle lugar a él. Esta cantidad viene dada por simple combinatoria:

$$\frac{3!}{j_{Gli}!j_{Gli_3}!j_{Gli_3R}!j_0!}.$$

Tenemos por tanto, la expresión de todas las configuraciones de equilibrio sin el papel del ARNp:

$$\begin{aligned} Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli_3R}, j_P = 0; C) = \\ = C(C) \frac{3!}{j_{Gli}!j_{Gli_3}!j_{Gli_3R}!j_0!} [B] \left(\frac{[Gli]}{K_{Gli}} \right)^{j_{Gli}} \left(\frac{[Gli_3]}{K_{Gli_3}} \right)^{j_{Gli_3}} \left(\frac{[Gli_3R]}{K_{Gli_3R}} \right)^{j_{Gli_3R}}. \end{aligned} \quad (3.7)$$

Supongamos ahora que el ARN está en nuestro complejo (esto es, $j_p = 1$). De nuevo vamos a tener unos espacios ocupados y otros vacíos, dados por: $j_{Gli}, j_{Gli_3}, j_{Gli_3R}, j_0$.

Todas las formas posibles configuraciones para obtener una concentración de equilibrio con $j_{Gli}, j_{Gli_3}, j_{Gli_3R}, j_P$ activadores, represores de nuevo se obtienen con la misma expresión. Sin embargo, ahora tenemos en cuenta el valor que aporta el ARNp. Siguiendo lo expuesto en los preliminares, la existencia de activadores y represores, modifica la afinidad de unión del ARN. Por consiguiente, el factor del ARNp dentro de nuestro complejo, en ausencia de factores viene expresado como:

$$[B \cdot ARNp] = \frac{[ARNp]}{K_{RP}} [B]. \quad (3.8)$$

En caso de que existan factores de transcripción tenemos que modificar la afinidad según comentábamos:

$$\frac{K_{RP}}{a_{Gli}^{j_{Gli}} a_{Gli_3}^{j_{Gli_3}} r_{Gli_3R}^{j_{Gli_3R}}}. \quad (3.9)$$

Donde $a_{Gli} a_{Gli_3} r_{Gli_3R}$ son las constante de cooperatividad y anticooperatividad asociadas a los factores de transcripción (y aparecen en la expresión modificadas según el numero de cada una, lo cual viene indicado en el exponente)

Tenemos pues la expresión para describir el estado:

$$C(C) \frac{3!}{j_{Gli}!j_{Gli_3}!j_{Gli_3R}!j_0!} [B] \frac{[RNAP] a_{Gli}^{j_{Gli}} a_{Gli_3}^{j_{Gli_3}} r_{Gli_3R}^{j_{Gli_3R}}}{K_{RP}} \left(\frac{[Gli]}{K_{Gli}} \right)^{j_{Gli}} \left(\frac{[Gli_3]}{K_{Gli_3}} \right)^{j_{Gli_3}} \left(\frac{[Gli_3R]}{K_{Gli_3R}} \right)^{j_{Gli_3R}}. \quad (3.10)$$

Agrupando por exponentes, podemos obtener la expresión final:

$$\begin{aligned}
 Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P = 1; C) = \\
 = C(C) \frac{3!}{j_{Gli}! j_{Gli_3}! j_{Gli3R}! j_0!} [B] \frac{[RNAP]}{K_{RP}} \left(\frac{a_{gli}[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{a_{gli_3}[Gli_3]}{K_{Gli_3}} \right)^{j_{gli_3}} \left(\frac{r_{gli3R}[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}}.
 \end{aligned} \tag{3.11}$$

Esto nos permite describir todo el espacio muestral, es decir, el espacio de todas las posibles configuraciones, atendiendo al numero de factores de transcripción y a la existencia o no de ARN polimerasa:

$$\Omega = \{(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P); j_{Gli}, j_{Gli_3}, j_{Gli3R} \geq 0; j_{Gli} + j_{Gli_3} + j_{Gli3R} \leq 3, j_P = 0, 1\}. \tag{3.12}$$

Definición de la probabilidad de cada configuración

Una vez que hemos descrito todas las configuraciones posibles en términos de las concentraciones de activador Gli y Gli3, del represor Gli3R y ARN polimerasa, obtenemos fácilmente la probabilidad de encontrar el promotor en una configuración particular de j_P ARN polimerasa y de factores de transcripción $j_{Gli}, j_{Gli_3}, j_{Gli3R}$ relacionados por una cooperatividad c . Sencillamente, asumimos que la probabilidad de cierto estado se va a obtener dividiendo las configuraciones que dan lugar a él, entre todas las posibles.

$$P^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P; C) = \frac{Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P; C)}{\sum_{\{j'_{Gli}, j'_{Gli_3}, j'_{Gli3R}, j'_P\} \in \Omega} Z^{(3)}(j'_{Gli}, j'_{Gli_3}, j'_{Gli3R}, j'_P; C)}, \tag{3.13}$$

con $(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P) \in \Omega$.

Resultado final del operador BEWARE

Como último paso obtenemos el operador BEWARE. Para ello debemos tener en cuenta que, según el enfoque que estamos siguiendo (recruitment) en [Shea and Ackers, 1985] se pone en manifiesto la necesidad de el anclaje de ARNp para que se lleve a cabo el proceso de transcripción, por tanto, la síntesis de proteína se producirá en los estados en los que $j_P = 1$ siguiendo el estudio de [Bintu et al., 2005, Frank et al., 2012].

Es decir, la expresión vendrá dada por:

$$\begin{aligned} BEWARE([Gli][Gli_3][Gli3R][ARNP]; C) &= \\ &= C_B \sum_{\substack{j_{Gli}+j_{Gli_3}+j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli_3}, j_{Gli3R} \geq 0}} P^{(n)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P = 1; C). \end{aligned} \quad (3.14)$$

Dividimos ahora el denominador en dos sumas, según la ARN polimerasa este ligada o no a la configuración:

$$\begin{aligned} BEWARE([Gli][Gli_3][Gli3R][ARNP]; C) &= \\ &= C_B \sum_{\substack{j_{Gli}+j_{Gli_3}+j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli_3}, j_{Gli3R} \geq 0}} \frac{Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P = 1; C)}{Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P = 0; C) + Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P = 1; C)}. \end{aligned} \quad (3.15)$$

Eso nos deja la expresión:

$$\begin{aligned} BEWARE([Gli][Gli_3][Gli3R][ARNP]; C) &= \\ &= \frac{C_B}{1 + \frac{\sum_{\substack{j_{Gli}+j_{Gli_3}+j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli_3}, j_{Gli3R} \geq 0}} Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P=0; C)}{\sum_{\substack{j_{Gli}+j_{Gli_3}+j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli_3}, j_{Gli3R} \geq 0}} Z^{(3)}(j_{Gli}, j_{Gli_3}, j_{Gli3R}, j_P=1; C)}}}. \end{aligned} \quad (3.16)$$

Esta expresión puede simplificarse si construimos lo que hemos denominado, función de regulación. En términos de la función del factor de regulación, F_{reg} , nuestro operador quedaría de la forma:

$$BEWARE(Gli, Gli_3, Gli3R) = \frac{C_b}{1 + \frac{k_{RNAP}}{F_{reg}(Gli, Gli_3, Gli3R)RNAP}}. \quad (3.17)$$

Para calcular cómo podemos expresar F_{reg} de una forma más compacta vamos a usar el teorema multinomial [Hall, 1998].

Teorema 3.2.1 (Teorema multinomial). *Para cualquier entero positivo m y cualquier entero no negativo n , la fórmula multinomial indica cómo una suma con m términos se expande cuando se eleva a una potencia arbitraria n :*

$$(x_1 + x_2 + \cdots + x_m)^n = \sum_{k_1+k_2+\cdots+k_m=n} \binom{n}{k_1, k_2, \dots, k_m} \prod_{1 \leq t \leq m} x_t^{k_t},$$

donde

$$\binom{n}{k_1, k_2, \dots, k_m} = \frac{n!}{k_1! k_2! \dots k_m!}$$

es un **coeficiente multinomial**.

En nuestro caso la expresión que vamos buscando obtener es de la forma:

$$(x_0 + x_{Gli} + x_{Gli3} + x_{Gli3R})^3 = \sum_{j_0 + j_{Gli} + j_{Gli3} + j_{Gli3R} = 3} \frac{3!}{j_0! j_{Gli}! j_{Gli3}! j_{Gli3R}!} x_0^{j_0} x_{Gli}^{j_{Gli}} x_{Gli3}^{j_{Gli3}} x_{Gli3R}^{j_{Gli3R}}. \quad (3.18)$$

Los sumatorios de nuestra expresión no coinciden exactamente con la expresión del teorema, así que operamos para convertirlos:

$$\begin{aligned} & \sum_{\substack{j_{Gli} + j_{Gli3} + j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli3}, j_{Gli3R} \geq 0}} C(C) \frac{3!}{j_{Gli}! j_{Gli3}! j_{Gli3R}! j_0!} \left(\frac{a_{gli}[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{a_{gli3}[Gli3]}{K_{Gli3}} \right)^{j_{gli3}} \left(\frac{r_{gli3R}[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}} \\ &= 1 + \sum_{\substack{j_{Gli} + j_{Gli3} + j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli3}, j_{Gli3R} \geq 1}} C(C) \frac{3!}{j_{Gli}! j_{Gli3}! j_{Gli3R}! j_0!} \left(\frac{a_{gli}[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{a_{gli3}[Gli3]}{K_{Gli3}} \right)^{j_{gli3}} \left(\frac{r_{gli3R}[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}} \\ &= 1 + \frac{1}{c} \sum_{\substack{j_{Gli} + j_{Gli3} + j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli3}, j_{Gli3R} \geq 1}} \frac{3!}{j_{Gli}! j_{Gli3}! j_{Gli3R}! j_0!} \left(\frac{ca_{gli}[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{ca_{gli3}[Gli3]}{K_{Gli3}} \right)^{j_{gli3}} \left(\frac{cr_{gli3R}[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}} \\ &= 1 - \frac{1}{c} + \frac{1}{c} \sum_{\substack{j_{Gli} + j_{Gli3} + j_{Gli3R} \leq n \\ j_{Gli}, j_{Gli3}, j_{Gli3R} \geq 0}} \frac{3!}{j_{Gli}! j_{Gli3}! j_{Gli3R}! j_0!} \left(\frac{ca_{gli}[Gli]}{K_{Gli}} \right)^{j_{gli}} \left(\frac{ca_{gli3}[Gli3]}{K_{Gli3}} \right)^{j_{gli3}} \\ & \cdot \left(\frac{cr_{gli3R}[Gli3R]}{K_{Gli3R}} \right)^{j_{gli3R}}. \end{aligned} \quad (3.19)$$

Estamos ahora en condiciones de utilizar el teorema 3.18 en el último paso de 3.19 para obtener la expresión:

$$\begin{aligned} & S^{(3)}(a_{Gli}[Gli]K_{Gli}^{-1}, a_{Gli3}[Gli3]K_{Gli3}^{-1}, r_{Gli3R}[Gli3R]K_{Gli3R}^{-1}; C) = \\ &= 1 - \frac{1}{c} + \frac{1}{c} \left(\frac{ca_{gli}[Gli]}{K_{Gli}} + \frac{ca_{gli3}[Gli3]}{K_{Gli3}} + \frac{cr_{gli3R}[Gli3R]}{K_{Gli3R}} \right)^3. \end{aligned} \quad (3.20)$$

Teniendo en cuenta, la expresión anterior, podemos describir la función de regulación

como:

$$F_{reg}([Gli][Gli_3][Gli3R]; C) = \frac{S^{(3)}(a_{Gli}[Gli]K_{Gli}^{-1}, a_{Gli_3}[Gli_3]K_{Gli_3}^{-1}, r_{Gli3R}[Gli3R]K_{Gli3R}^{-1}; C)}{S^{(3)}([Gli]K_{Gli}^{-1}, [Gli_3]K_{Gli_3}^{-1}, [Gli3R]K_{Gli3R}^{-1}; C)}. \quad (3.21)$$

Concluimos así el cálculo del modelado del operador BEWARE con un enfoque *recruitment*. Con este operador y las ecuaciones definidas en el capítulo 1 podemos disponer ya de nuestro modelo completo.

3.3. Sistema final

La mayoría de cuentas del apartado se han generado con la ayuda de [Meurer et al., 2017].

$$\frac{dGli}{dt} = BEWARE(Gli, Gli_3, Gli3R) - k_{deg}Gli, \quad (3.22)$$

$$\frac{dGli_3}{dt} = \frac{r_{g3b}}{Ptc} - Gli_3 \left(k_{deg} + \frac{k_{g3rc}}{K_{g3rc} + Signal} \right), \quad (3.23)$$

$$\frac{dGli3R}{dt} = Gli_3 \left(\frac{k_{g3rc}}{K_{g3rc} + Signal} \right) - k_{deg}Gli3R, \quad (3.24)$$

$$\frac{dPtc}{dt} = BEWARE(Gli, Gli_3, Gli3R) - k_{degp}Ptc. \quad (3.25)$$

Donde tenemos, por definición:

$$Signal = \frac{\frac{Shh}{k_{shh}} + 1}{\frac{Shh}{k_{shh}} + 1 + \frac{Ptc}{k_{ptc}}}, \quad (3.26)$$

y,

$$BEWARE(Gli, Gli_3, Gli3R) = \frac{c_b}{1 + \frac{k_{RNAP}}{F_{reg}(Gli, Gli_3, Gli3R)RNAP}}}, \quad (3.27)$$

donde solo nos queda describir F_{reg} . En el caso de de gradientes opuestos y no/total

cooperatividad de los factores de transcripción nos queda:

$$F_{reg} = \frac{1 + \frac{1}{c} \left(\frac{Gli a_{Gli}}{k_{Gli}} c + \frac{Gli_3 a_{Gli_3}}{k_{Gli_3 R}} c + \frac{Gli_3 R c}{k_{Gli_3 R}} r_{Gli_3 R} + 1 \right)^3 - \frac{1}{c}}{1 + \frac{1}{c} \left(\frac{Gli c}{k_{Gli}} + \frac{Gli_3 c}{k_{Gli_3 R}} + \frac{Gli_3 R c}{k_{Gli_3 R}} + 1 \right)^3 - \frac{1}{c}}. \quad (3.28)$$

Podemos desarrollar las funciones en cada uno de los términos, quedándonos las siguientes expresiones:

$$\frac{dGli}{dt} = -Gli k_{deg} + \frac{C_b}{1 + \frac{k_{RNAP} \left(1 + \frac{1}{c} \left(\frac{Gli c}{k_{Gli}} + \frac{Gli_3 c}{k_{Gli_3 R}} + \frac{Gli_3 R c}{k_{Gli_3 R}} + 1 \right)^3 - \frac{1}{c} \right)}{RNAP \left(1 + \frac{1}{c} \left(\frac{Gli a_{Gli}}{k_{Gli}} c + \frac{Gli_3 a_{Gli_3}}{k_{Gli_3 R}} c + \frac{Gli_3 R c}{k_{Gli_3 R}} r_{Gli_3 R} + 1 \right)^3 - \frac{1}{c} \right)}}. \quad (3.29)$$

$$\frac{dGli_3}{dt} = -Gli_3 \left(k_{deg} + \frac{k_{g3rc}}{K_{g3rc} + \frac{\frac{Shh}{k_{Shh}} + 1}{\frac{Shh}{k_{Shh}} + 1 + \frac{ptc}{k_{ptc}}}} \right) + \frac{r_{g3b}}{ptc}. \quad (3.30)$$

$$\frac{dGli_3 R}{dt} = Gli_3 \left(-Gli_3 R k_{deg} + \frac{k_{g3rc}}{K_{g3rc} + \frac{\frac{Shh}{k_{Shh}} + 1}{\frac{Shh}{k_{Shh}} + 1 + \frac{ptc}{k_{ptc}}}} \right). \quad (3.31)$$

$$\frac{dPtc}{dt} = \frac{C_b}{1 + \frac{k_{RNAP} \left(1 + \frac{1}{c} \left(\frac{Gli c}{k_{Gli}} + \frac{Gli_3 c}{k_{Gli_3 R}} + \frac{Gli_3 R c}{k_{Gli_3 R}} + 1 \right)^3 - \frac{1}{c} \right)}{RNAP \left(1 + \frac{1}{c} \left(\frac{Gli a_{Gli}}{k_{Gli}} c + \frac{Gli_3 a_{Gli_3}}{k_{Gli_3 R}} c + \frac{Gli_3 R c}{k_{Gli_3 R}} r_{Gli_3 R} + 1 \right)^3 - \frac{1}{c} \right)}} - k_{deg p} Ptc. \quad (3.32)$$

3.4. Estados estacionarios

Siguiendo con el estudio estándar que se lleva a cabo en los modelos matemáticos procedemos con un estudio sobre los estados estacionarios que podemos encontrar en nuestro modelo. En primer lugar procedemos afrontando el problema desde una perspectiva analítica.

Sean las ecuaciones (3.22), (3.23), (3.24), (3.25), si suponemos que éstas se encuentran en un estado estacionario entonces sus cantidades son constantes. Esto implica que su derivada temporal es igual a cero.

El procedimiento transcurre exactamente igual que en el segundo capítulo. La única diferencia será la expresión del BEWARE una vez que sustituyamos el valor dependiente de Gli en el resto de variables. Por ello, nos centramos en las simulaciones del apartado siguiente.

3.5. Simulaciones

En esta sección vamos a desarrollar las simulaciones numéricas llevadas a cabo en el estudio cualitativo del modelo que acabamos de presentar.

Para estudiar este modelo y reproducir algunos resultados contrastando los mismo con el estudio del modelo clásico, seguimos el mismo esquema:

- **Recolección y contraste de los parámetros** usados, basándonos en tanto el modelo clásico como en el planteamiento de este tipo de modelado en [Cambon and Sanchez, 2017].
- **Análisis numérico de las soluciones estacionarias:** Desarrollamos la fórmula analítica para obtener un código que nos permita rastrear cambios en el comportamiento cualitativo ante grandes variaciones en los parámetros.
- **Estudio y comparación de la variabilidad del operador BEWARE.** Dentro de este apartado comparamos numéricamente el comportamiento de nuestro operador, con el observado por el modelo clásico, alcanzó una similitud notable.

3.5.1. Parámetros

Salvo que especifiquemos lo contrario, los valores de los parámetros serán los recogidos en la tabla 3.1

3.5.2. Variación del operador BEWARE

Como avanzábamos hemos querido estudiar como se comporta cualitativamente el operador BEWARE en las mismas condiciones que el operador BEWARE clásico.

| Tabla de parámetros, operador <i>BEWARE</i> | | | |
|---|------------------------|--|----------------------------|
| Parámetro | Valor | Descripción | Fuente |
| c | 1 | Constante positiva (valor 1 implica cooperatividad total) | [Cambon and Sanchez, 2017] |
| a_{Gli} | 4.35 | Intensidad de represión transcripcional de Gli | [Cambon and Sanchez, 2017] |
| a_{Gli3} | 4.35 | Intensidad de represión transcripcional de Gli3 | [Cambon and Sanchez, 2017] |
| r_{Gli3R} | 5×10^{-5} | Intensidad de represión transcripcional de Gli | [Cambon and Sanchez, 2017] |
| k_{Gli} | 9×10^1 | Constante de disociación de los activadores para los potenciadores genéticos | [Cambon and Sanchez, 2017] |
| k_{Gli3} | 9×10^1 | Constante de disociación de los activadores para los potenciadores genéticos | [Cambon and Sanchez, 2017] |
| k_{Gli3R} | 9×10^1 | Constante de disociación de los represores para los potenciadores genéticos | [Cambon and Sanchez, 2017] |
| k_{RNAP} | 1 | Afinidad de unión de RNA polimerasa | [Cambon and Sanchez, 2017] |
| $RNAP$ | 1 | Concentración de RNA polimerasa | [Cambon and Sanchez, 2017] |
| c_b | 1 nMmin^{-1} | Constante del operador | [Cambon and Sanchez, 2017] |

Cuadro 3.1: Tabla de parámetros, operador *BEWARE*

Dentro las figuras se puede apreciar una línea discontinua. Esta línea marca el comportamiento asintótico del nuevo operador BEWARE. Podemos observar que si bien el comportamiento y la tendencia es similar, el nuevo operador ofrece un resultado mucho mas continuo en un gran rango de parámetros Figura 3.2 así como una mayor suavidad en el comportamiento en los rangos más usado por [Lai et al., 2004], Figura 3.1.

3.5.3. Evolución temporal

Realizando un estudio similar al del primer modelo, observamos un inamovible estado estacionario (cuyo nivel hemos hecho coincidir con el del modelo clásico a través de los parámetros libres, C_B). La existencia de un punto fijo dentro de nuestro marco de parámetros y valores de la variable no parecía alterarse: fueran cuales fueran las condiciones iniciales y los parámetros impuestos sólo podíamos obtener un estado estable.

El análisis de bifurcaciones se muestra infructuoso, puesto que solo arrojaba una rama

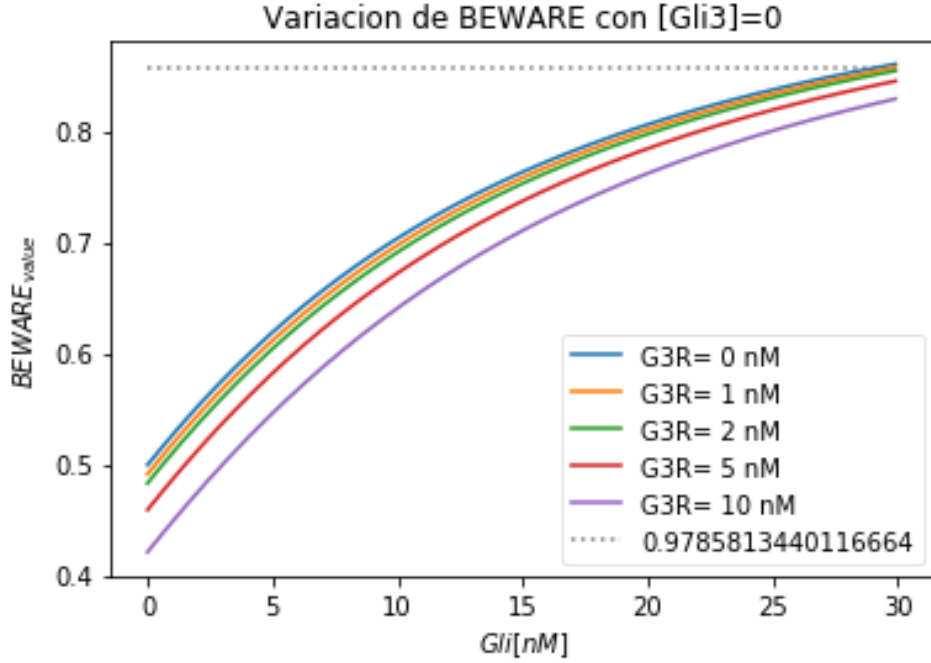


Figura 3.1: Variación del nuevo operador BEWARE

estable, así que esto nos hizo pensar en ampliar de forma masiva nuestros parámetros en busca de observar si este comportamiento era global.

En la figura 3.3 se puede observar el comportamiento al cual nos referimos. Muy similar al caso con Shh/k_{Shh} del modelo clásico.

3.5.4. Análisis numérico de los estados estacionarios

Siguiendo el mismo esquema del programa para localizar ceros y contarlos usado en el modelo anterior ² hemos rastreado el comportamiento de nuestro modelo en amplios márgenes.

El resultado es siempre el mismo, nuestro modelo conserva un punto de corte (En la figura 3.4 mostramos el comportamiento más habitual). Esto nos induce a pensar que en nuestro marco de trabajo, una prueba teórica de este comportamiento es posible, si bien durante el trabajo esta prueba ha mostrado ser un problema difícil, debido a la complejidad de las expresiones.

²El programa se diseñó en un primer momento para atacar este problema, pero se utilizó en el modelo anterior para comprobar su fiabilidad.

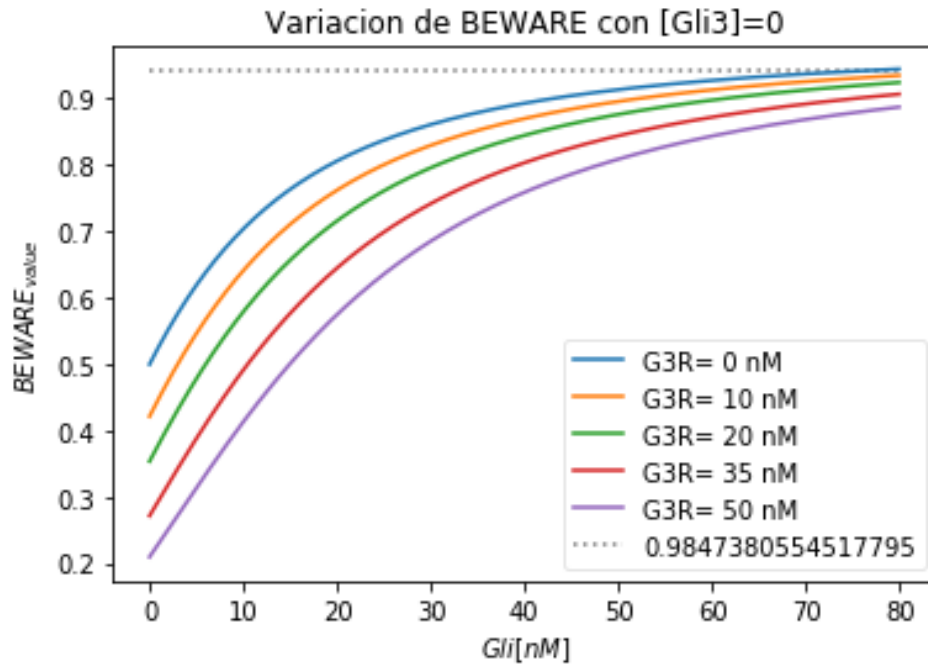


Figura 3.2: Variación del nuevo operador BEWARE en más rango

Exponemos finalmente una muestra del output del programa en 3.5, donde se observa el comportamiento que venimos observando de manera general.

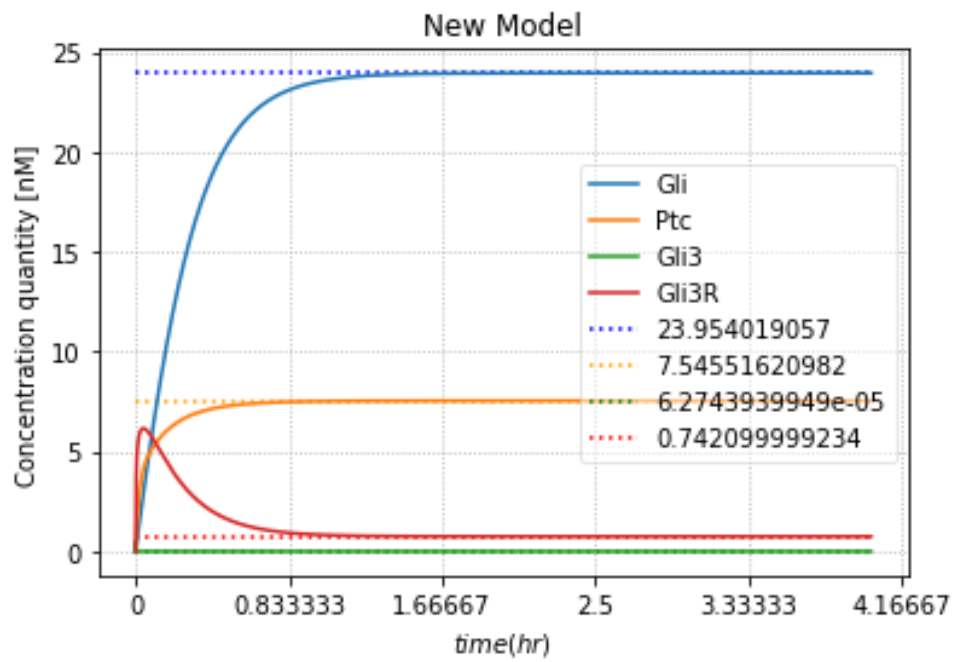


Figura 3.3: Evolución temporal del nuevo operador BEWARE

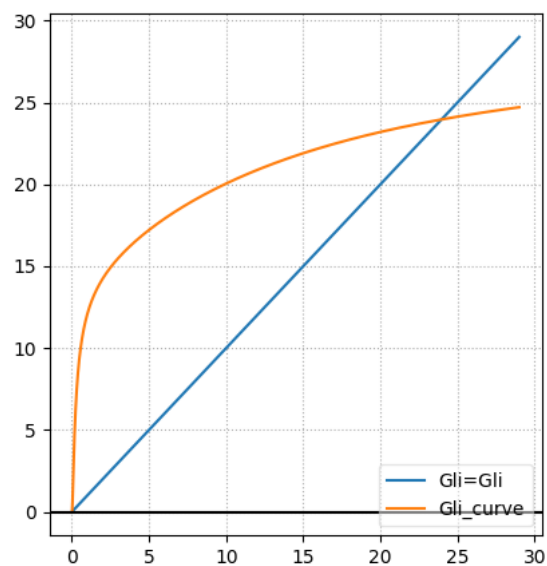


Figura 3.4: Representación gráfica de las rectas involucradas en la ecuación de punto fijo para el nuevo BEWARE

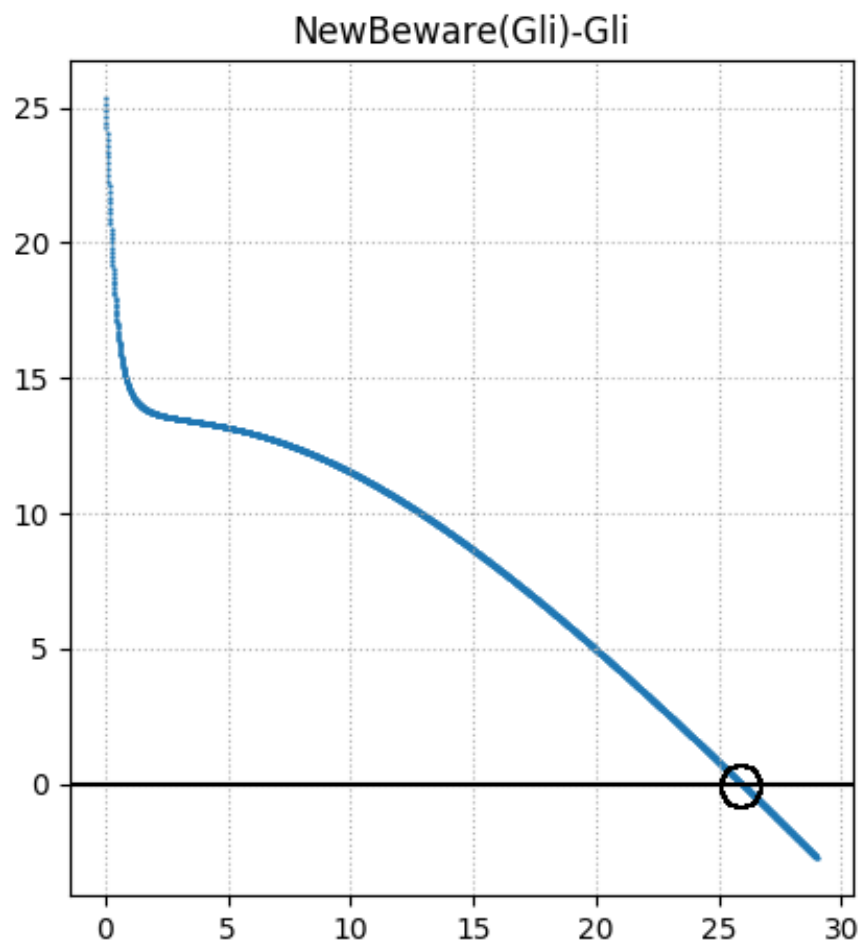


Figura 3.5: Representación gráfica del algoritmo de localización de ceros, que busca en qué puntos corta la recta generada por la resta entre los miembros de la ecuación de punto fijo generada con el nuevo BEWARE al eje de abscisas

Capítulo 4

Conclusiones

4.1. Conclusiones y trabajo futuro

1. Hemos presentado en nuestro trabajo un estudio de dos modelos utilizados para entender el sistema de señalización de Shh.
2. Dentro del modelo clásico hemos observado comportamientos cualitativos y como su origen puede residir también en otros parámetros no contemplados en el artículo original.
3. A su vez, hemos presentado un nuevo modelado para este problema siguiendo una estrategia distinta a la original.
4. Este modelo ha demostrado comportarse de forma parecida al original, pero con sutiles diferencia en cuanto a la estabilidad. En concreto este modelo no presenta un swith biestable en su comportamiento en ninguna simulación numérica realizada hasta la fecha. Este hecho apoya la teoría de que este modelo capta de forma correcta la realidad biológica observada en la práctica.
5. Nos gustaría seguir desarrollando este proyecto. Si bien el numérico a demostrado ser una interesante herramienta para captar tendencias cualitativas en nuestro modelo, necesitamos una solida prueba teórica.
6. Para ello, a parte de utilizar barridos de parámetros mayores, queremos investigar si podemos obtener una prueba teórica de este resultado. Iniciamos este trabajo con un fin teórico, sin embargo la complejidad de las operaciones resiste nuestros

envites de resolución actuales. Aún así, las pruebas numéricas sostienen que vamos por buen camino.

Bibliografía

- [Ay and Arnosti, 2011] Ay, A. and Arnosti, D. N. (2011). Mathematical modeling of gene expression: a guide for the perplexed biologist. *Critical reviews in biochemistry and molecular biology*, 46(2):137–151.
- [Bintu et al., 2005] Bintu, L., Buchler, N. E., Garcia, H. G., Gerland, U., Hwa, T., Kondev, J., and Phillips, R. (2005). Transcriptional regulation by the numbers: models. *Current opinion in genetics & development*, 15(2):116–124.
- [Cambon and Sanchez,] Cambon, M. and Sanchez, O. Beware modules with multiple competitive transcription factors, work in progress. Work in progress.
- [Cambon and Sanchez, 2017] Cambon, M. and Sanchez, O. (2017). Analysis of biochemical mechanisms provoking differential spatial expression in Hh target genes. *ArXiv e-prints*.
- [Chen et al., 1999] Chen, C.-H., von Kessler, D. P., Park, W., Wang, B., Ma, Y., and Beachy, P. A. (1999). Nuclear trafficking of cubitus interruptus in the transcriptional regulation of hedgehog target gene expression. *Cell*, 98(3):305–316.
- [Clement et al., 2007] Clement, V., Sanchez, P., De Tribolet, N., Radovanovic, I., and i Altaba, A. R. (2007). Hedgehog-gli1 signaling regulates human glioma growth, cancer stem cell self-renewal, and tumorigenicity. *Current biology*, 17(2):165–172.
- [Dahmane et al., 1997] Dahmane, N., Lee, J., Robins, P., Heller, P., and i Altaba, A. R. (1997). Activation of the transcription factor gli1 and the sonic hedgehog signalling pathway in skin tumours. *Nature*, 389(6653):876.
- [Frank et al., 2012] Frank, T. D., Carmody, A. M., and Kholodenko, B. N. (2012). Versatility of cooperative transcriptional activation: a thermodynamical modeling analysis for greater-than-additive and less-than-additive effects. *PloS one*, 7(4):e34439.

- [French and Lauffenburger, 1996] French, A. R. and Lauffenburger, D. A. (1996). Intracellular receptor/ligand sorting based on endosomal retention components. *Biotechnology and bioengineering*, 51(3):281–297.
- [Gilbert, 2000] Gilbert, S. (2000). Developmental biology. 6th edition.
- [Hall, 1998] Hall, M. (1998). *Combinatorial theory*, volume 71. John Wiley & Sons.
- [Hornbeck PV, 2015] Hornbeck PV, Zhang B, M. B. K. J. L. V. S. E. (2015). Phosphositeplus, 2014: mutations, ptms and recalibrations. *nucleic acids res.* 43:D512-20.
- [Lai et al., 2004] Lai, K., Robertson, M. J., and Schaffer, D. V. (2004). The sonic hedgehog signaling system as a bistable genetic switch. *Biophysical Journal*, 86(5):2748–2757.
- [Meijer et al., 2012] Meijer, H., Dercole, F., and Oldeman, B. (2012). Numerical bifurcation analysis. In *Mathematics of Complexity and Dynamical Systems*, pages 1172–1194. Springer.
- [Meurer et al., 2017] Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roučka, v., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., and Scopatz, A. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.
- [Ortiz, 2017] Ortiz, B. (2017). Análisis cualitativo de sistemas dinámicos con origen biológico. Mathematics bachelor’s degree thesis, Universidad de Granada.
- [Ruiz i Altaba, 1999] Ruiz i Altaba, A. (1999). Gli proteins and hedgehog signaling: development and cancer. *Trends in genetics*, 15(10):418–425.
- [Saha and Schaffer, 2006] Saha, K. and Schaffer, D. V. (2006). Signal dynamics in sonic hedgehog tissue patterning. *Development*, 133(5):889–900.
- [Scatchard, 1949] Scatchard, G. (1949). The attractions of proteins for small molecules and ions. *Annals of the New York Academy of Sciences*, 51(4):660–672.
- [Shea and Ackers, 1985] Shea, M. A. and Ackers, G. K. (1985). The or control system of bacteriophage lambda: A physical-chemical model for gene regulation. *Journal of molecular biology*, 181(2):211–230.

- [Taipale et al., 2002] Taipale, J., Cooper, M., Maiti, T., and Beachy, P. (2002). Patched acts catalytically to suppress the activity of smoothened. *Nature*, 418(6900):892.
- [Wang et al., 2000] Wang, B., Fallon, J. F., and Beachy, P. A. (2000). Hedgehog-regulated processing of gli3 produces an anterior/posterior repressor gradient in the developing vertebrate limb. *Cell*, 100(4):423–434.
- [Wikipedia contributors, 2018a] Wikipedia contributors (2018a). Lac operon — Wikipedia, the free encyclopedia. [Online; accessed 16-July-2018].
- [Wikipedia contributors, 2018b] Wikipedia contributors (2018b). Sonic hedgehog — Wikipedia, the free encyclopedia. [Online; accessed 16-July-2018].
- [Wikipedia contributors, 2018c] Wikipedia contributors (2018c). Transcription factors — Wikipedia, the free encyclopedia. [Online; accessed 16-July-2018].

Anexo A

Códigos

Los códigos también se pueden consultar en <https://github.com/theboooort/shh-signal-model>

A.1. Python

Variabilidad del nuevo operador BEWARE

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Tue Jun 26 21:39:40 2018
5
6 @author: boooort
7 """
8
9
10 from math import *
11 import scipy as sp
12 from scipy.integrate import odeint
13 import matplotlib.pyplot as plt
14 import matplotlib.ticker as ticker
15 # Definition of constants
16
```

```

17 # from beware new model:
18
19 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
      1 anti-cooperativity
20 a_Gli = 4.35 # transcriptional activation intensity for Gli
21 a_Gli3 = 4.35 # transcriptional activation intensity for Gli
22 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
23 k_Gli = 9*10**1 # dissociation constant of activators for Gene enhancers
24 k_Gli3 = 9*10**1 # dissociation constant of activators for Gene enhancers
25 k_Gli3R = 9*10**1 # dissociation constant of repressors for Gene
      enhancers
26 k_RNAP = 1 # RNA polymerase binding affinity
27 RNAP = 1 # RNA polymerase concentration
28 c_b = 1 # BEWARE constant
29
30 # from Lai-Schaffer classic model
31
32 Shh = 0 # Shh quantity [0,30]
33 k_shh = 0.58 # dissociation constant shh-Ptc bindings [0.58,2.0]
34 k_Ptc = 8.3*10**-2 # half maximal concentration of Ptc which inhibits smo
      signlaing
35 k_deg = 0.009 # degradation constant for all Gli related proteins
36 k_g3rc = 0.012 # rate constant for the conversion to signal strenGh
37 r_g3b = 1.6*10**-1 # basal rate of Gli3 synthesis
38 K_g3rc = 0.1 # sensitivity constant of the conversion to signal strenGh
39 k_deg_p = 0.09 # degradation rate constant for Ptc [0.045,0.071]
40
41
42 # regulation function with non/total cooperativity
43
44 def F_reg_nt_coop(Gli, Gli3, Gli3R):
45
46     return (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+r_Gli3R*c*
      Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*Gli3R/
      k_Gli3R)**3)
47
48 # Signal function
49

```

```

50
51 def Signal(Ptc):
52
53     return (1+Shh/k_shh)/(1+Shh/k_shh+Ptc/k_Ptc)
54
55 # Beware operator
56
57
58 def BEWARE(Gli, Gli3, Gli3R):
59
60     return c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop(Gli, Gli3, Gli3R)))
61
62
63 # Frist we define our range and values
64 Gli = sp.arange(0.0, 80.0, 0.1)
65 Gli3 = 0
66 Gli3R_values = [0, 10, 20, 35, 50]
67
68 # Plotting configuration
69 fig, ax = plt.subplots()
70 ax.set_ylabel(r"$BEWARE_{value}$")
71 ax.set_xlabel(r'$Gli[nM]$')
72 plt.title(r'Variacion de BEWARE con [Gli3]=0')
73 for Gli3R in Gli3R_values:
74     ax.plot(Gli, BEWARE(Gli, Gli3, Gli3R), label='G3R= '+str(Gli3R)+' nM')
75 ax.hlines(y=BEWARE(len(Gli)/10-1, 0, 0), xmin=0, xmax=len(Gli)/10,
76         linewidth=1.5, color='grey', linestyle='dotted', label=str(BEWARE(len(
77         Gli)-1, 0, 0)))
76 ax.legend(loc='lower right', fancybox=True, framealpha=0.5)
77 plt.show()

```

Variabilidad de los operadores promotor y basal de [Saha and Schaffer, 2006].

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Mon Jul  2 09:52:05 2018
5
6 @author: booort

```

```

7  """
8
9
10 from math import *
11 import scipy as sp
12 from scipy.integrate import odeint
13 import matplotlib.pyplot as plt
14 import matplotlib.ticker as ticker
15 # Definition of constants
16
17 # from beware new model:
18
19 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
      1 anti-cooperativity
20 a_Gli = 4.35 # transcriptional activation intensity for Gli
21 a_Gli3 = 4.35 # transcriptional activation intensity for Gli
22 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
23 k_Gli = 9*10**1 # dissociation constant of activators for Gene enhancers
24 k_Gli3 = 9*10**1 # dissociation constant of activators for Gene enhancers
25 k_Gli3R = 9*10**1 # dissociation constant of repressors for Gene
      enhancers
26 k_RNAP = 1 # RNA polymerase binding affinity
27 RNAP = 1 # RNA polymerase concentration
28 c_b = 1 # BEWARE constant
29
30 # from Lai-Schaffer classic model
31
32 Shh = 0 # Shh quantity [0,30]
33 k_shh = 0.58 # dissociation constant shh-Ptc bindings [0.58,2.0]
34 k_Ptc = 8.3*10**-2 # half maximal concentration of Ptc which inhibits smo
      signlaing
35 k_deg = 0.009 # degradation constant for all Gli related proteins
36 k_g3rc = 0.012 # rate constant for the conversion to signal strenGh
37 r_g3b = 1.6*10**-1 # basal rate of Gli3 synthesis
38 K_g3rc = 0.1 # sensitivity constant of the conversion to signal strenGh
39 k_deg_p = 0.09 # degradation rate constant for Ptc [0.045,0.071]
40 # -----
41 K1 = 8.3*10**-1

```

```

42 K2 = 8.3*10**-1
43 c = 1
44 e = 0.5
45 r = 0.2
46 v_max = 2.4*10**-1
47 r_bas = v_max/100
48 v_maxp = 7.5*10**-1
49 r_basp = v_maxp/100
50 # regulation function with non/total cooperativity
51
52 def F_reg_nt_coop(Gli, Gli3, Gli3R):
53
54     return (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+r_Gli3R*c*
55             Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*Gli3R/
56             k_Gli3R)**3)
57
58 # Signal function
59
60 def Signal(Ptc):
61
62     return (1+Shh/k_shh)/(1+Shh/k_shh+Ptc/k_Ptc)
63
64 # Beware operator
65
66 def BEWARE(Gli, Gli3, Gli3R):
67
68     return c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop(Gli, Gli3, Gli3R)))
69
70 def Promoter(Gli, Gli3, Gli3R):
71
72     return ((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+Gli*
73             K2+2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1*K2*
74             r + 3*e**2*Gli3R**2*K1**2*r**2 + Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r))))
75     / (3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*K1
76             + Gli*K2)**3 +K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))

```



```

74 def Basal(Gli, Gli3, Gli3R):
75     return (3*c*K1*K2*(Gli3*K1+ Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 +
        Gli*K2 + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2+ 3*
        Gli3R*r)))/ (3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1
        + Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli +
        K1)*K2))
76
77
78 # Frist we define our range and values
79 Gli = sp.arange(0.0, 30.0, 0.1)
80 Gli3 = 0
81 Gli3R_values = [0, 1, 2, 5, 10]
82
83 # Plotting configuration
84 fig, ax = plt.subplots()
85 ax.set_ylabel(r"$BEWARE_{value}$")
86 ax.set_xlabel(r'$Gli[nM]$',)
87 plt.title(r'Variacion de BEWARE con [Gli3]=0')
88 for Gli3R in Gli3R_values:
89     ax.plot(Gli, BEWARE(Gli, Gli3, Gli3R), label='G3R= '+str(Gli3R)+' nM')
90 ax.hlines(y=BEWARE(len(Gli)/10-1, 0, 0), xmin=0, xmax=len(Gli)/10,
        linewidth=1.5, color='grey', linestyle='dotted', label=str(BEWARE(len(
        Gli)-1, 0, 0)))
91 ax.legend(loc='lower right', fancybox=True, framealpha=0.5)
92 plt.show()
93
94 # Plotting configuration
95 fig, ax = plt.subplots()
96 ax.set_ylabel(r"$Promoter_{value}$")
97 ax.set_xlabel(r'$Gli[nM]$',)
98 plt.title(r'Variacion de Promoter con [Gli3]=0')
99 for Gli3R in Gli3R_values:
100     ax.plot(Gli, Promoter(Gli, Gli3, Gli3R), label='G3R= '+str(Gli3R)+' nM'
        )
101 ax.hlines(y=BEWARE(len(Gli)/10-1, 0, 0), xmin=0, xmax=len(Gli)/10,
        linewidth=1.5, color='grey', linestyle='dotted', label=str(BEWARE(len(
        Gli)-1, 0, 0)))
102 ax.legend(loc='lower right', fancybox=True, framealpha=0.5)

```

```

103 plt.show()
104
105 fig, ax = plt.subplots()
106 ax.set_ylabel(r"$Basal_{value}$")
107 ax.set_xlabel(r"$Gli[nM]$"')
108 plt.title(r'Variacion de Basal con [Gli3]=0')
109 for Gli3R in Gli3R_values:
110     ax.plot(Gli, Basal(Gli, Gli3, Gli3R), label='G3R= '+str(Gli3R)+' nM')
111 ax.hlines(y=BEWARE(len(Gli)/10-1, 0, 0), xmin=0, xmax=len(Gli)/10,
112         linewidth=1.5, color='grey', linestyle='dotted', label=str(BEWARE(len(
113     Gli)-1, 0, 0)))
112 ax.legend(loc='lower right', fancybox=True, framealpha=0.5)
113 plt.show()

```

Comparativa entre las expresiones de BEWARE.

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Sun Jul  1 21:38:51 2018
5
6 @author: booort
7 """
8
9
10 import scipy as sp
11 from scipy.integrate import odeint
12 import matplotlib.pyplot as plt
13 import matplotlib.ticker as ticker
14 # Definition of constants
15
16 # from beware new model:
17
18 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
19       1 anti-cooperativity
20 a_Gli = 4.35 # transcriptional activation intensity for Gli
21 a_Gli3 = 4.35 # transcriptional activation intensity for Gli
22 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
23 k_Gli = 90 # dissociation constant of activators for Gene enhancers

```

```

23 k_Gli3 = 90 # dissociation constant of activators for Gene enhancers
24 k_Gli3R = 90 # dissociation constant of repressors for Gene enhancers
25 k_RNAP = 1 # RNA polymerase binding affinity
26 RNAP = 1 # RNA polymerase concentration
27 c_b = 0.26 # BEWARE constant
28 c_b1 = 3.15
29 # from Lai-Schaffer classic model
30
31 Shh = 15 # Shh quantity [0,30]
32 k_shh = 0.58 # dissociation constant shh-ptc bindings [0.58,2.0]
33 k_ptc = 8.3*10**-2 # 1/2maximal concentration of ptc which inhibits smo
    signlaing
34 k_deg = 0.009 # degradatation constant for all Gli related proteins
35 k_g3rc = 0.012 # rate constant for the conversion to signal strenGh
36 r_g3b = 1.6*10**-1 # basal rate of Gli3 synthesis
37 K_g3rc = 0.1 # sensitivity constant of the conversion to signal strenGh
38 k_deg_p = 0.09 # degradation rate constant for Ptc [0.045,0.071]
39 # -----
40 K1 = 8.3*10**-1
41 K2 = 8.3*10**-1
42
43 e = 0.5
44 r = 0.2
45 v_max = 2.4*10**-1
46 r_bas = v_max/100
47 v_maxp = 7.5*10**-1
48 r_basp = v_maxp/100
49
50
51 def Signal(Ptc):
52
53     return (1+Shh/k_shh)/(1+Shh/k_shh+Ptc/k_ptc)
54
55
56 def Promoter(Gli, Gli3, Gli3R):
57     return ((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+Gli*
        K2+2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1*K2*
        r + 3*e**2*Gli3R**2*K1**2*r**2 + Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r))))

```

```

/ (3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*K1
+ Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))
58
59
60 def Basal(Gli, Gli3, Gli3R):
61     return (3*c*K1*K2*(Gli3*K1+ Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 +
Gli*K2 + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2+ 3*
Gli3R*r)))/ (3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1
+ Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli +
K1)*K2))
62
63
64 def lai_saha_model(X, t):
65     Gli, Gli3, Gli3R, Ptc = X
66
67     dGli_dt = v_max*Promoter(Gli, Gli3, Gli3R)+r_bas*Basal(Gli, Gli3, Gli3R
)-k_deg*Gli
68     dGli3_dt = r_g3b/Gli-Gli3*(k_deg+k_g3rc/(K_g3rc+Signal(Ptc)))
69     dGli3R_dt = Gli3*(k_g3rc/(K_g3rc+Signal(Ptc)))-k_deg*Gli3R
70     dPtc_dt = v_maxp*Promoter(Gli, Gli3, Gli3R)+r_basp*Basal(Gli, Gli3,
Gli3R)-k_deg_p*Ptc
71
72     return dGli_dt, dGli3_dt, dGli3R_dt, dPtc_dt
73
74
75 # Frist we define our temporal range
76 t = sp.arange(0.0, 1200.0, 0.1)
77
78 # definition of odeint for solve the system numerically
79
80 vector_solution = odeint(lai_saha_model, [0.01, 0, 0, 0], t)
81
82 # Extraction of Gli,gli3,gli3r,ptc numerical values of the solution
83
84 evol_gli = vector_solution[:, 0]
85 evol_ptc = vector_solution[:, 3]
86 evol_gli3 = vector_solution[:, 1]
87 evol_gli3r = vector_solution[:, 2]

```

```

88 # regulation function with non/total cooperativity
89
90
91 def F_reg_nt_coop(Gli, Gli3, Gli3R):
92
93     return (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+r_Gli3R*c*
94         Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*Gli3R/
95         k_Gli3R)**3)
96
97
98 def BEWARE(Gli, Gli3, Gli3R):
99
100     return c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop(Gli, Gli3, Gli3R)))
101
102
103 def shh_evolution_system(X, t):
104     Gli, Gli3, Gli3R, Ptc = X
105
106     dGli_dt = BEWARE(Gli, Gli3, Gli3R)-k_deg*Gli
107     dGli3_dt = r_g3b/Gli-Gli3*(k_deg+k_g3rc/(K_g3rc+Signal(Ptc)))
108     dGli3R_dt = Gli3*(k_g3rc/(K_g3rc+Signal(Ptc)))-k_deg*Gli3R
109     dPtc_dt = c_b1*BEWARE(Gli, Gli3, Gli3R)-k_deg_p*Ptc
110
111     return dGli_dt, dGli3_dt, dGli3R_dt, dPtc_dt
112
113 # Frist we define our temporal range
114 t = sp.arange(0.0, 1200.0, 0.1)
115
116 # definition of odeint for solve the system numerically
117
118 vector_solution = odeint(shh_evolution_system, [0.01, 0, 0, 0], t)
119
120 # Extraction of Gli,gli3,gli3r,ptc numerical values of the solution
121
122 evol_gli_1 = vector_solution[:, 0]
123 evol_ptc_1 = vector_solution[:, 3]
124 evol_gli3_1 = vector_solution[:, 1]

```

```

124 evol_gli3r_1 = vector_solution[:, 2]
125
126 # Plotting the results (scaling them previously)
127
128 fig, ax = plt.subplots()
129 ax.plot(t, evol_gli, label=r'Gli Lai-Saha',color='purple' )
130 ax.plot(t, evol_gli_1, label=r'Gli New Model',color='orange')
131 plt.title('Gli Lai-Saha vs. Gli New Model')
132 scale_x = 60
133 ticks_x = ticker.FuncFormatter(lambda x, pos: '{0:g}'.format(x/scale_x))
134 ax.xaxis.set_major_formatter(ticks_x)
135 ax.set_xlabel(r"$time(hr)$")
136 ax.set_ylabel(r'Concentration quantity [nM]')
137 ax.hlines(y=evol_gli_1[-1], xmin=0, xmax=len(evol_gli_1)/10, linewidth=1.5
138         ,color='orange', linestyle='dotted', label=str(evol_gli_1[-1]))
139 ax.hlines(y=evol_gli[-1], xmin=0, xmax=len(evol_gli)/10, linewidth=1.5 ,
140         color='purple', linestyle='dotted', label=str(evol_gli[-1]))
141 ax.legend(loc='best', fancybox=True, framealpha=0.5)
142 plt.show()
143
144 fig, ax = plt.subplots()
145 ax.plot(t, evol_ptc, label=r'Ptc Lai-Saha',color='purple' )
146 ax.plot(t, evol_ptc_1, label=r'Ptc New Model',color='orange' )
147 plt.title('Ptc Lai-Saha vs. Ptc New Model')
148 scale_x = 60
149 ticks_x = ticker.FuncFormatter(lambda x, pos: '{0:g}'.format(x/scale_x))
150 ax.xaxis.set_major_formatter(ticks_x)
151 ax.set_xlabel(r"$time(hr)$")
152 ax.set_ylabel(r'Concentration quantity [nM]')
153 ax.hlines(y=evol_ptc_1[-1], xmin=0, xmax=len(evol_ptc_1)/10, linewidth=1.5
154         ,color='orange', linestyle='dotted', label=str(evol_ptc_1[-1]))
155 ax.hlines(y=evol_ptc[-1], xmin=0, xmax=len(evol_ptc)/10, linewidth=1.5 ,
156         color='purple', linestyle='dotted', label=str(evol_ptc[-1]))
157 ax.legend(loc='best', fancybox=True, framealpha=0.5)
158 plt.show()
159
160 fig, ax = plt.subplots()
161 ax.plot(t, evol_gli3, label=r'Gli3 Lai-Saha',color='purple' )

```

```

158 ax.plot(t, evol_gli3_1, label=r'Gli3 New Model',color='orange' )
159 plt.title('Gli3 Lai-Saha vs. Gli3 New Model')
160 scale_x = 60
161 ticks_x = ticker.FuncFormatter(lambda x, pos: '{0:g}'.format(x/scale_x))
162 ax.xaxis.set_major_formatter(ticks_x)
163 ax.set_xlabel(r"$time(hr)$")
164 ax.set_ylabel(r'Concentration quantity [nM]')
165 ax.hlines(y=evol_gli3_1[-1], xmin=0, xmax=len(evol_gli3_1)/10, linewidth
           =1.5 ,color='orange', linestyle='dotted', label=str(evol_gli3_1[-1]))
166 ax.hlines(y=evol_gli3[-1], xmin=0, xmax=len(evol_gli3)/10, linewidth=1.5 ,
           color='purple', linestyle='dotted', label=str(evol_gli3[-1]))
167 ax.legend(loc='best', fancybox=True, framealpha=0.5)
168 plt.show()
169
170 fig, ax = plt.subplots()
171 ax.plot(t, evol_gli3r, label=r'Gli3R Lai-Saha',color='purple' )
172 ax.plot(t, evol_gli3r_1, label=r'Gli3R New Model',color='orange' )
173 plt.title('Gli3R Lai-Saha vs. Gli3R New Model')
174 scale_x = 60
175 ticks_x = ticker.FuncFormatter(lambda x, pos: '{0:g}'.format(x/scale_x))
176 ax.xaxis.set_major_formatter(ticks_x)
177 ax.set_xlabel(r"$time(hr)$")
178 ax.set_ylabel(r'Concentration quantity [nM]')
179 ax.hlines(y=evol_gli3r_1[-1], xmin=0, xmax=len(evol_gli3r_1)/10, linewidth
           =1.5 ,color='orange', linestyle='dotted', label=str(evol_gli3r_1[-1]))
180 ax.hlines(y=evol_gli3r[-1], xmin=0, xmax=len(evol_gli3r)/10, linewidth=1.5
           ,color='purple', linestyle='dotted', label=str(evol_gli3r[-1]))
181 ax.legend(loc='best', fancybox=True, framealpha=0.5)
182 plt.show()
183
184 print(evol_gli_1[-1],evol_gli[-1],evol_gli_1[-1]-evol_gli[-1])
185 print(evol_ptc_1[-1],evol_ptc[-1],evol_ptc_1[-1]-evol_ptc[-1])
186 print(evol_gli3_1[-1],evol_gli3[-1],evol_gli3_1[-1]-evol_gli3[-1])
187 print(evol_gli3r_1[-1],evol_gli3r[-1],evol_gli3r_1[-1]-evol_gli3r[-1])

```

Script para contar los ceros.

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-

```

```

3 """
4 Created on Tue Jul 10 22:13:08 2018
5
6 @author: booort
7 """
8
9 #!/usr/bin/env python3
10 # -*- coding: utf-8 -*-
11 """
12 Created on Tue Jul 10 18:37:30 2018
13
14 @author: booort
15 """
16
17 import scipy as sp
18 import numpy as np
19 from scipy.integrate import odeint
20 import matplotlib.pyplot as plt
21 import matplotlib.ticker as ticker
22 import logging
23 LOG_FILENAME = 'Output.log'
24 logging.basicConfig(filename=LOG_FILENAME,level=logging.DEBUG)
25 file = open('resultado.txt','w')
26
27 def count_zeros(vector):
28     count = 0
29     for i in range(len(vector)-2):
30         if vector[i] > 0 and vector[i+1] < 0 :
31             count += 1
32         elif vector[i] < 0 and vector[i+1] > 0 :
33             count += 1
34     return count
35
36
37 def gli_curve(Gli,vector):
38     Shh, k_shh, k_ptc, k_deg, k_g3rc, r_g3b, K_g3rc, k_deg_p, K1, K2, c, e,
39     r, v_max, r_bas, v_maxp, r_basp, k_cc = vector
40     Ptc = k_cc*Gli

```



```

40     Signal = (1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
41     Gli3 = (r_g3b*(K_g3rc+Signal))/((k_deg*(K_g3rc+Signal)+k_g3rc)*Gli)
42     Gli3R = (r_g3b)/(k_deg*Gli)-Gli3
43     Basal = (3*c*K1*K2*(Gli3*K1+ Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 +
    Gli*K2 + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2+ 3*
    Gli3R*r)))/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1
    + Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli +
    K1)*K2))
44     Promoter = ((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+
    Gli*K2+2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1
    *K2*r + 3*e**2*Gli3R**2*K1**2*r**2 + Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r)
    )))/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*
    K1 + Gli*K2)**3 +K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))
45     return v_max*(Promoter+0.01*Basal)/k_deg
46
47
48 def gli_curve_1(Gli,vector):
49     Shh, k_shh, k_ptc, k_deg, k_g3rc, r_g3b, K_g3rc, k_deg_p, c, a_Gli,
    a_Gli3, r_Gli3R, k_Gli, k_Gli3, k_Gli3R, k_RNAP, RNAP, c_b, c_b1 =
    vector
50     Ptc = (0.89*60*k_deg)/(k_deg_p*c_b)*Gli
51     Signal = (1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
52     Gli3 = (r_g3b*(K_g3rc+Signal))/((k_deg*(K_g3rc+Signal)+k_g3rc)*Gli)
53     Gli3R = (r_g3b)/(k_deg*Gli)-Gli3
54     F_reg_nt_coop = (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+
    r_Gli3R*c*Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*
    Gli3R/k_Gli3R)**3)
55     beware = c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop))
56     return (beware/k_deg)
57
58
59 def muestreo(value,longitude):
60     vector=[]
61     for i in range(1,longitude,1):
62         if (value-2/i) > 0:
63             vector.append(value-2/i)
64     vector.append(value)
65     for i in range(-longitude,0,1):

```

```

66         vector.append(value-2/i)
67     return vector
68
69
70 def variability_2_by_2_lai_saha(param1,param2):
71     variability_vector_1=muestreo(parameters[param1],3)
72     variability_vector_2=muestreo(parameters[param2],3)
73     for i in variability_vector_1:
74         for j in variability_vector_2:
75             parameters_aux = parameters.copy()
76             parameters_aux[param1]=i
77             parameters_aux[param2]=j
78             logging.debug('{}={}, {}={}, Zeros: {}'.format(
parameters_name[param1],i,parameters_name[param2],j,count_zeros(
gli_curve(Gli,parameters_aux)-Gli))
79             if count_zeros(gli_curve(Gli,parameters_aux)-Gli)==1:
80                 file.write('{}={}, {}={}, Zeros: {} \n'.format(
parameters_name[param1],i,parameters_name[param2],j,count_zeros(
gli_curve(Gli,parameters_aux)-Gli))
81
82 def variability_2_by_2_new_beware(param1,param2):
83     variability_vector_1=muestreo(parameters2[param1],8)
84     variability_vector_2=muestreo(parameters2[param2],8)
85     for i in variability_vector_1:
86         for j in variability_vector_2:
87             parameters_aux = parameters2.copy()
88             parameters_aux[param1]=i
89             parameters_aux[param2]=j
90             logging.debug('{}={}, {}={}, Zeros: {}'.format(
parameters2_name[param1],i,parameters2_name[param2],j,count_zeros(
gli_curve_1(Gli,parameters_aux)-Gli))
91             if count_zeros(gli_curve_1(Gli,parameters_aux)-Gli)==3 or
count_zeros(gli_curve_1(Gli,parameters_aux)-Gli)==2 :
92                 file.write('{}={}, {}={}, Zeros: {} \n'.format(
parameters2_name[param1],i,parameters2_name[param2],j,count_zeros(
gli_curve_1(Gli,parameters_aux)-Gli))
93
94 # Definition of constants

```

```

95 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
    1 anti-cooperativity
96
97 a_Gli = 4.35 # transcriptional activation intensity for Gli
98 a_Gli3 = 4.35 # transcriptional activation intensity for Gli3
99 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
100
101 k_Gli = 9*10**1 # dissociation constant of activators for Gene enhancers
102 k_Gli3 = 9*10**1 # dissociation constant of activators for Gene enhancers
103 k_Gli3R = 9*10**1 # dissociation constant of repressors for Gene
    enhancers
104
105 k_RNAP = 1 # RNA polymerase binding affinity
106 RNAP = 1 # RNA polymerase concentration
107
108 c_b = 0.26*60 # BEWARE constant
109 c_b1 = 3.15*60
110
111
112 # from Lai-Schaffer classic model shared with BEWARE
113
114 Shh = 0.1*1 # Shh quantity [0,30]
115 k_shh = 1 # dissociation constant shh-ptc bindings [0.58,2.0]
116
117 k_ptc = 8.3*10**-2 # 1/2maximal concentration of ptc which inhibits smo
    signlaing
118 k_deg = 0.54 # degradatation constant for all Gli related proteins
119
120 k_g3rc = 0.012*60 # rate constant for the conversion to signal strenGh
121 r_g3b = 60*0.16 # basal rate of Gli3 synthesis
122
123 K_g3rc = 0.1*10**0 # sensitivity constant of the conversion to signal
    strenGh
124 k_deg_p = 0.09*60 # degradatation rate constant for Ptc [0.045,0.071]
125 # Exclusively from lai-saha
126 K1 = 8.3*10**-1
127 K2 = 8.3*10**-1
128 c = 1

```

```

129 e = 0.5
130 r = 0.2
131 v_max = 60*2.4*10**-1
132 r_bas = v_max/100
133 v_maxp = 60*7.5*10**-1
134 r_basp = v_maxp/100
135 k_cc = (v_maxp*k_deg)/(v_max*k_deg_p)
136
137 parameters = [Shh, k_shh, k_ptc, k_deg, k_g3rc, r_g3b, K_g3rc, k_deg_p, K1,
               K2, c, e, r, v_max, r_bas, v_maxp, r_basp, k_cc]
138 parameters_name = ['Shh', 'k_shh', 'k_ptc', 'k_deg', 'k_g3rc', 'r_g3b', '
                    K_g3rc', 'k_deg_p', 'K1', 'K2', 'c', 'e', 'r', 'v_max', 'r_bas', 'v_maxp',
                    'r_basp', 'k_cc']
139
140 parameters2 = [Shh, k_shh, k_ptc, k_deg, k_g3rc, r_g3b, K_g3rc, k_deg_p, c,
                a_Gli, a_Gli3, r_Gli3R, k_Gli, k_Gli3, k_Gli3R, k_RNAP, RNAP, c_b, c_b1
                ]
141 parameters2_name = ['Shh', 'k_shh', 'k_ptc', 'k_deg', 'k_g3rc', 'r_g3b', '
                    K_g3rc', 'k_deg_p', 'c', 'a_Gli', 'a_Gli3', 'r_Gli3R', 'k_Gli', 'k_Gli3',
                    'k_Gli3R', 'k_RNAP', 'RNAP', 'c_b', 'c_b1']
142
143 mesh_size=0.001
144 Gli = sp.arange(0.01, 29.0, mesh_size)
145
146 for parametro_1 in range(len(parameters2)):
147     for position in range(len(parameters2)):
148         print(position)
149         variability_2_by_2_new_beware(parametro_1-1,position)
150 file.close()

```

Definición y simulación del sistema con el nuevo operador BEWARE.

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Tue Jun 12 17:48:12 2018
5
6 This script is created to define all the functions needed in my study
7

```

```

8 @author: booort
9 """
10
11 import scipy as sp
12 from scipy.integrate import odeint
13 import matplotlib.pyplot as plt
14 import matplotlib.ticker as ticker
15 # Definition of constants
16
17 # from beware new model:
18
19 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
      1 anti-cooperativity
20 a_Gli = 4.35 # transcriptional activation intensity for Gli
21 a_Gli3 = 4.35 # transcriptional activation intensity for Gli
22 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
23 k_Gli = 9*10**1 # dissociation constant of activators for Gene enhancers
24 k_Gli3 = 9*10**1 # dissociation constant of activators for Gene enhancers
25 k_Gli3R = 9*10**1 # dissociation constant of repressors for Gene
      enhancers
26 k_RNAP = 1 # RNA polymerase binding affinity
27 RNAP = 1 # RNA polymerase concentration
28 c_b = 0.26 # BEWARE constant
29 c_b1 = 3.15
30 # from Lai-Schaffer classic model
31
32 Shh = 0.1 # Shh quantity [0,30]
33 k_shh = 0.58 # dissociation constant shh-ptc bindings [0.58,2.0]
34 k_ptc = 8.3*10**-2 # half maximal concentration of ptc which inhibits smo
      signlaing
35 k_deg = 0.009 # degradation constant for all Gli related proteins
36 k_g3rc = 0.012*1000 # rate constant for the conversion to signal strenGh
37 r_g3b = 1.6*10**-1 # basal rate of Gli3 synthesis
38 K_g3rc = 0.1 # sensitivity constant of the conversion to signal strenGh
39 k_deg_p = 0.09 # degradation rate constant for Ptc [0.045,0.071]
40
41
42 # regulation function with non/total cooperativity

```

```

43
44 def F_reg_nt_coop(Gli, Gli3, Gli3R):
45
46     return (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+r_Gli3R*c*
Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*Gli3R/
k_Gli3R)**3)
47
48
49 def Signal(Ptc):
50
51     return (1+Shh/k_shh)/(1+Shh/k_shh+Ptc/k_ptc)
52
53
54 def BEWARE(Gli, Gli3, Gli3R):
55
56     return c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop(Gli, Gli3, Gli3R)))
57
58
59 def shh_evolution_system(X, t):
60     Gli, Gli3, Gli3R, Ptc = X
61
62     dGli_dt = BEWARE(Gli, Gli3, Gli3R)-k_deg*Gli
63     dGli3_dt = r_g3b/Gli-Gli3*(k_deg+k_g3rc/(K_g3rc+Signal(Ptc)))
64     dGli3R_dt = Gli3*(k_g3rc/(K_g3rc+Signal(Ptc)))-k_deg*Gli3R
65     dPtc_dt = c_b1*BEWARE(Gli, Gli3, Gli3R)-k_deg_p*Ptc
66
67     return dGli_dt, dGli3_dt, dGli3R_dt, dPtc_dt
68
69
70 # Frist we define our temporal range
71 t = sp.arange(0.0, 2400.0, 0.1)
72
73 # definition of odeint for solve the system numerically
74
75 vector_solution = odeint(shh_evolution_system, [0.01, 0, 0, 0], t)
76
77 # Extraction of Gli,gli3,gli3r,ptc numerical values of the solution
78

```

```

79 evol_gli_1 = vector_solution[:, 0]
80 evol_ptc_1 = vector_solution[:, 3]
81 evol_gli3_1 = vector_solution[:, 1]
82 evol_gli3r_1 = vector_solution[:, 2]
83
84 # Plotting the results (scaling them previously)
85 fig, ax = plt.subplots()
86 ax.plot(t, evol_gli_1, label=r'Gli' )
87 ax.plot(t, evol_ptc_1, label=r'Ptc' )
88 ax.plot(t, evol_gli3_1, label=r'Gli3' )
89 ax.plot(t, evol_gli3r_1, label=r'Gli3R' )
90 scale_x = 600
91 ticks_x = ticker.FuncFormatter(lambda x, pos: '{0:g}'.format(x/scale_x))
92 ax.xaxis.set_major_formatter(ticks_x)
93
94 ax.set_xlabel(r"$time(hr)$")
95 ax.set_ylabel(r'Concentration quantity [nM]')
96 ax.hlines(y=evol_gli_1[-1], xmin=0, xmax=len(evol_gli_1)/10, linewidth=1.5
97         ,color='blue', linestyle='dotted', label=str(evol_gli_1[-1]))
98 ax.hlines(y=evol_ptc_1[-1], xmin=0, xmax=len(evol_ptc_1)/10, linewidth=1.5
99         ,color='orange', linestyle='dotted', label=str(evol_ptc_1[-1]))
100 ax.hlines(y=evol_gli3_1[-1], xmin=0, xmax=len(evol_gli3_1)/10, linewidth
101         =1.5 ,color='green', linestyle='dotted', label=str(evol_gli3_1[-1]))
102 ax.hlines(y=evol_gli3r_1[-1], xmin=0, xmax=len(evol_gli3r_1)/10, linewidth
103         =1.5 ,color='red', linestyle='dotted', label=str(evol_gli3r_1[-1]))
104 ax.legend(loc='right', fancybox=True, framealpha=0.5)
105 plt.title('New Model')
106 ax.grid(True, which='both', ls=':')
107 plt.show()
108
109 print(evol_gli_1[2000])

```

Scrip para representar la ecuacion de punto fijo.

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Tue Jul  3 20:36:54 2018
5

```

```

6 @author: booort
7 """
8
9
10 import scipy as sp
11 from scipy.integrate import odeint
12 import matplotlib.pyplot as plt
13 import matplotlib.ticker as ticker
14 # Definition of constants
15 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
      1 anti-cooperativity
16
17 a_Gli = 4.35 # transcriptional activation intensity for Gli
18 a_Gli3 = 4.35 # transcriptional activation intensity for Gli3
19 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
20
21 k_Gli = 9*10**1 # dissociation constant of activators for Gene enhancers
22 k_Gli3 = 9*10**4 # dissociation constant of activators for Gene enhancers
23 k_Gli3R = 9*10**1 # dissociation constant of repressors for Gene
      enhancers
24
25 k_RNAP = 1 # RNA polymerase binding affinity
26 RNAP = 1 # RNA polymerase concentration
27
28 c_b = 0.26*60 # BEWARE constant
29 c_b1 = 3.15*60
30 # from Lai-Schaffer classic model
31
32 Shh = 0.1*1 # Shh quantity [0,30]
33 k_shh = 1 # dissociation constant shh-ptc bindings [0.58,2.0]
34
35 k_ptc = 8.3*10**-2 # 1/2maximal concentration of ptc which inhibits smo
      signlaing
36 k_deg = 0.009*60 # degradation constant for all Gli related proteins
37
38 k_g3rc = 0.012*60 # rate constant for the conversion to signal strenGh
39 r_g3b = 60*0.16 # basal rate of Gli3 synthesis
40

```



```

41 K_g3rc = 0.1*10**0 # sensitivity constant of the conversion to signal
    strenGh
42 k_deg_p = 0.09*60 # degradation rate constant for Ptc [0.045,0.071]
43 # -----
44 K1 = 8.3*10**-1
45 K2 = 8.3*10**-1
46 c = 1
47 e = 0.5
48 r = 0.2
49 v_max = 60*2.4*10**-1
50 r_bas = v_max/100
51 v_maxp = 60*7.5*10**-1
52 r_basp = v_maxp/100
53 k_cc = (v_maxp*k_deg)/(v_max*k_deg_p)
54
55 def gli_curve(Gli):
56     Ptc = k_cc*Gli
57     Signal = (1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
58     Gli3 = (r_g3b*(K_g3rc+Signal))/((k_deg*(K_g3rc+Signal)+k_g3rc)*Gli)
59     Gli3R = (r_g3b)/(k_deg*Gli)-Gli3
60     Basal = (3*c*K1*K2*(Gli3*K1+ Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 +
        Gli*K2 + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2+ 3*
        Gli3R*r)))/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1
        + Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli +
        K1)*K2))
61     Promoter = ((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+
        Gli*K2+2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1
        *K2*r + 3*e**2*Gli3R**2*K1**2*r**2 + Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r)
        )))/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*
        K1 + Gli*K2)**3 +K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))
62     return v_max*(Promoter+0.01*Basal)/k_deg
63
64 def gli_curve_1(Gli):
65     Ptc = (0.89*60*k_deg)/(k_deg_p*c_b)*Gli
66     Signal = (1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
67     Gli3 = (r_g3b*(K_g3rc+Signal))/((k_deg*(K_g3rc+Signal)+k_g3rc)*Gli)
68     Gli3R = (r_g3b)/(k_deg*Gli)-Gli3
69     F_reg_nt_coop = (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+

```

```

    r_Gli3R*c*Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*
    Gli3R/k_Gli3R)**3)
70     beware = c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop))
71     return (beware/k_deg)
72
73 mesh_size=0.001
74 Gli = sp.arange(0.01, 29.0, mesh_size)
75 Gli_1 = sp.arange(0.01, 29.0, mesh_size)
76 fig = plt.figure()
77 ax=fig.add_subplot(1,2,1)
78 ax.plot(Gli_1, Gli_1,label='Gli=Gli')
79 ax.plot(Gli, gli_curve(Gli), label='Gli_curve')
80 ax.grid(True, which='both',ls=':')
81 ax.axhline(y=0, color='k')
82 # plt.axvline(ymin=0,ymax=30, x=0.899264058328, linewidth=1.5 ,color='black
    ', label='24')
83 # plt.axvline(ymin=0,ymax=30, x=23.7805433472, linewidth=1.5 ,color='black
    ', label='24')
84 ax.legend(loc='lower right', fancybox=True, framealpha=0.5)
85
86
87 mesh_size=0.001
88 Gli = sp.arange(0.01, 29.0, mesh_size)
89 Gli_1 = sp.arange(0.01, 29.0, mesh_size)
90 ax=fig.add_subplot(1,2,2)
91 ax.plot(Gli_1,Gli_1,label='Gli=Gli')
92 ax.plot(Gli, gli_curve_1(Gli), label='Gli_curve')
93 ax.grid(True, which='both',ls=':')
94 ax.axhline(y=0, color='k')
95 # plt.axvline(ymin=0,ymax=30, x=24, linewidth=1.5 ,color='black', label
    ='24')
96 ax.legend(loc='lower right', fancybox=True, framealpha=0.5)
97 plt.show()

```

Definición y simulación del sistema con el operador BEWARE de [Saha and Schaffer, 2006].

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-

```

```

3  """
4  Created on Sun Jul  1 21:01:45 2018
5
6  @author: booort
7  """
8
9  import scipy as sp
10 from scipy.integrate import odeint
11 import matplotlib.pyplot as plt
12 import matplotlib.ticker as ticker
13 # Definition of constants
14 # from Lai-Schaffer classic model
15
16 Shh = 0.1 # Shh quantity [0,30]
17 #Shh = 1.5 # Uncomment to change parameter
18 k_shh = 1 # dissociation constant shh-ptc bindings [0.58,2.0]
19 k_ptc = 8.3*10**-2 # 1/2maximal concentration of ptc which inhibits smo
    signlaing
20 k_deg = 0.009 # degradatation constant for all Gli related proteins
21 k_g3rc = 0.012 # rate constant for the conversion to signal strenGh
22 r_g3b = 1.6*10**-1 # basal rate of Gli3 synthesis
23 K_g3rc = 0.1 # sensitivity constant of the conversion to signal strenGh
24 k_deg_p = 0.09 # degradatation rate constant for Ptc [0.045,0.071]
25 # -----
26 K1 = 8.3*10**-1
27 K2 = 8.3*10**-1
28 c = 1
29 e = 0.5
30 r = 0.2
31 v_max = 2.4*10**-1
32 r_bas = v_max/100
33 v_maxp = 7.5*10**-1
34 r_basp = v_maxp/100
35
36
37 def Signal(Ptc):
38
39     return (1+(Shh/k_shh)) / (1+(Shh/k_shh)+(Ptc/k_ptc))

```

```

40
41
42 def Promoter(Gli, Gli3, Gli3R):
43     return ((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+Gli*
K2+2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1*K2*
r + 3*e**2*Gli3R**2*K1**2*r**2 + Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r)))
/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*K1
+ Gli*K2)**3 +K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))
44
45
46 def Basal(Gli, Gli3, Gli3R):
47     return (3*c*K1*K2*(Gli3*K1+ Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 +
Gli*K2 + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2+ 3*
Gli3R*r)))/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1
+ Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli +
K1)*K2))
48
49
50 def lai_saha_model(X, t):
51     Gli, Gli3, Gli3R, Ptc = X
52
53     dGli_dt = v_max*Promoter(Gli, Gli3, Gli3R)+r_bas*Basal(Gli, Gli3, Gli3R
)-k_deg*Gli
54
55     dGli3_dt = r_g3b/Gli-Gli3*(k_deg+(k_g3rc/(K_g3rc+Signal(Ptc))))
56
57     dGli3R_dt = Gli3*(k_g3rc/(K_g3rc+Signal(Ptc)))-k_deg*Gli3R
58
59     dPtc_dt = v_maxp*Promoter(Gli, Gli3, Gli3R)+r_basp*Basal(Gli, Gli3,
Gli3R)-k_deg_p*Ptc
60
61     return dGli_dt, dGli3_dt, dGli3R_dt, dPtc_dt
62
63
64 # Frist we define our temporal range
65 mesh_size=0.001
66 t = sp.arange(0.0, 4800.0, mesh_size)
67

```

```

68 # definition of odeint for solve the system numerically
69
70 #vector_solution = odeint(lai_saha_model, [1,0,0,0], t)
71 # vector_solution = odeint(lai_saha_model,
    [0.899269420907,4.06580533612,15.7032823132,0.281021254959], t)
72 vector_solution = odeint(lai_saha_model, [0.01,0,0,0], t)
73
74 # Extraction of Gli,gli3,gli3r,ptc numerical values of the solution
75
76 evol_gli = vector_solution[:, 0]
77 evol_ptc = vector_solution[:, 3]
78 evol_gli3 = vector_solution[:, 1]
79 evol_gli3r = vector_solution[:, 2]
80
81 # Plotting the results (scaling them previously)
82 fig, ax = plt.subplots()
83 ax.plot(t, evol_gli, label=r'Gli' )
84 ax.plot(t, evol_ptc, label=r'Ptc' )
85 ax.plot(t, evol_gli3, label=r'Gli3' )
86 ax.plot(t, evol_gli3r, label=r'Gli3R' )
87 scale_x = 60
88 ticks_x = ticker.FuncFormatter(lambda x, pos: '{0:g}'.format(x/scale_x))
89 ax.xaxis.set_major_formatter(ticks_x)
90
91 ax.set_xlabel(r"$time(hr)$")
92 ax.set_ylabel(r'Concentration quantity [nM]')
93 ax.hlines(y=evol_gli[-1], xmin=0, xmax=len(evol_gli)/1000, linewidth=1.5 ,
    color='blue', linestyle='dotted', label=str(evol_gli[-1]))
94 ax.hlines(y=evol_ptc[-1], xmin=0, xmax=len(evol_ptc)/1000, linewidth=1.5 ,
    color='orange', linestyle='dotted', label=str(evol_ptc[-1]))
95 ax.hlines(y=evol_gli3[-1], xmin=0, xmax=len(evol_gli3)/1000, linewidth=1.5
    ,color='green', linestyle='dotted', label=str(evol_gli3[-1]))
96 ax.hlines(y=evol_gli3r[-1], xmin=0, xmax=len(evol_gli3r)/1000, linewidth
    =1.5 ,color='red', linestyle='dotted', label=str(evol_gli3r[-1]))
97 #plt.axvline(ymin=0,ymax=6, x=30, linewidth=1.5 ,color='black', label='0.5
    h')
98 ax.legend(loc='right', fancybox=True, framealpha=0.5)
99 plt.title(r'Lai-Saha Model. $Shh/k\_Shh=$'+str(Shh))

```

```

100 plt.show()
101
102 print(evol_gli[-1])
103 print(evol_gli3[-1])
104 print(evol_gli3r[-1])
105 print(evol_ptc[-1])
106 print(Signal)

```

Script para localizar los ceros por las veces que la resta de dos expresiones corta el eje de abscisas.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Tue Jul 10 18:37:30 2018
5
6  @author: booort
7  """
8
9  import scipy as sp
10 import numpy as np
11 from scipy.integrate import odeint
12 import matplotlib.pyplot as plt
13 import matplotlib.ticker as ticker
14 import time
15 start_time = time.time()
16 # Definition of constants
17 c = 1 # positive constant, Greater than 1 implies cooperativity, less than
      1 anti-cooperativity
18
19 a_Gli = 4.35 # transcriptional activation intensity for Gli
20 a_Gli3 = 4.35 # transcriptional activation intensity for Gli3
21 r_Gli3R = 5*10**-5 # transcriptional repression intensity for Gli
22
23 k_Gli = 9*10**1 # dissociation constant of activators for Gene enhancers
24 k_Gli3 = 9*10**1 # dissociation constant of activators for Gene enhancers
25 k_Gli3R = 9*10**1 # dissociation constant of repressors for Gene
      enhancers
26

```

```

27 k_RNAP = 1 # RNA polymerase binding affinity
28 RNAP = 1 # RNA polymerase concentration
29
30 c_b = 0.26*60 # BEWARE constant
31 c_b1 = 3.15*60
32 # from Lai-Schaffer classic model shared with BEWARE
33
34 Shh = 1.1*1 # Shh quantity [0,30]
35 k_shh = 1 # dissociation constant shh-ptc bindings [0.58,2.0]
36
37 k_ptc = 8.3*10**-2 # 1/2maximal concentration of ptc which inhibits smo
    signlaing
38 k_deg = 0.009*60 # degradation constant for all Gli related proteins
39
40 k_g3rc = 0.012*60 # rate constant for the conversion to signal strenGh
41 r_g3b = 60*0.16 # basal rate of Gli3 synthesis
42
43 K_g3rc = 0.1*10**0 # sensitivity constant of the conversion to signal
    strenGh
44 k_deg_p = 0.09*60 # degradation rate constant for Ptc [0.045,0.071]
45 # Exclusively from lai-saha
46 K1 = 8.3*10**-1
47 K2 = 8.3*10**-1
48 c = 2
49 e = 0.5
50 r = 0.2
51 v_max = 60*2.4*10**-1
52 r_bas = v_max/100
53 v_maxp = 60*7.5*10**-1
54 r_basp = v_maxp/100
55 k_cc = (v_maxp*k_deg)/(v_max*k_deg_p)
56
57
58 def gli_curve(Gli):
59     Ptc = k_cc*Gli
60     Signal = (1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
61     Gli3 = (r_g3b*(K_g3rc+Signal))/((k_deg*(K_g3rc+Signal)+k_g3rc)*Gli)
62     Gli3R = (r_g3b)/(k_deg*Gli)-Gli3

```

```

63 Basal = (3*c*K1*K2*(Gli3*K1 + Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 +
    Gli*K2 + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2 +
    3*Gli3R*r))
64 ) / (3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*
    K1 + Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*
    Gli + K1)*K2))
65 Promoter = ((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+
    Gli*K2+2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1
    *K2*r + 3*e**2*Gli3R**2*K1**2*r**2 +
66
    Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r))))/(3*c*
    K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*K1 + Gli*
    K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))
67 return v_max*(Promoter+0.01*Basal)/k_deg
68
69
70 def gli_curve_1(Gli):
71     Ptc = (0.89*60*k_deg)/(k_deg_p*c_b)*Gli
72     Signal = (1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
73     Gli3 = (r_g3b*(K_g3rc+Signal))/((k_deg*(K_g3rc+Signal)+k_g3rc)*Gli)
74     Gli3R = (r_g3b)/(k_deg*Gli)-Gli3
75     F_reg_nt_coop = (1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+
    r_Gli3R*c *
76
    Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/
    k_Gli+c*Gli3/k_Gli3+c*Gli3R/k_Gli3R)**3)
77     beware = c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop))
78     return (beware/k_deg)
79
80
81 mesh_size = 0.1
82 Gli = sp.arange(0.01, 29, mesh_size)
83 Gli_1 = sp.arange(0.01, 29, mesh_size)
84 fig = plt.figure()
85 ax = fig.add_subplot(1, 2, 1)
86 plt.title('Lai-Saha(Gli)-Gli')
87 ax.scatter(Gli_1, gli_curve(Gli)-Gli_1, s=0.5)
88 ax.grid(True, which='both', ls=':')
89 ax.axhline(y=0, color='k')

```



```

90
91 mesh_size = 0.01
92 Gli = sp.arange(0.01, 29.0, mesh_size)
93 Gli_1 = sp.arange(0.01, 29.0, mesh_size)
94
95 ax = fig.add_subplot(1, 2, 2)
96 plt.title('NewBeware(Gli)-Gli')
97 ax.scatter(Gli_1, gli_curve_1(Gli)-Gli_1, s=0.5)
98 ax.grid(True, which='both', ls=':')
99 ax.axhline(y=0, color='k')
100 plt.show()
101
102
103 def count_zeros(vector):
104     count = 0
105     for i in range(len(vector)-2):
106         if vector[i] > 0 and vector[i+1] < 0:
107             count += 1
108         elif vector[i] < 0 and vector[i+1] > 0:
109             count += 1
110     return count
111
112
113 print(count_zeros(gli_curve(Gli)-Gli_1), count_zeros(gli_curve_1(Gli)-Gli_1
114             ))
115 print("--- %s seconds ---" % (time.time() - start_time))

```

Cáculo simbólico.

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Fri Jun 15 10:41:46 2018
5
6 this script is created to solve the steady states
7
8 @author: booort
9 """
10

```

```

11 from sympy import *
12 init_printing()
13 c = Symbol('c', real=True) # positive constant, greater than 1 implies
    cooperativity, less than 1 anti-cooperativity
14 a_gli = Symbol('a_gli', real=True) # transcriptional activation intensity
    for gli
15 a_gli3 = Symbol('a_gli3', real=True) # transcriptional activation
    intensity for gli
16 r_gli3R = Symbol('r_gli3R', real=True) # transcriptional repression
    intensity for gli
17 k_gli = Symbol('k_gli', real=True) # dissociation constant of activators
    for gene enhancers
18 k_gli3 = Symbol('k_gli3', real=True) # dissociation constant of
    activators for gene enhancers
19 k_gli3R = Symbol('k_gli3R', real=True) # dissociation constant of
    repressors for gene enhancers
20 k_RNAP = Symbol('k_RNAP', real=True) # RNA polymerase binding affinity
21 RNAP = Symbol('RNAP', real=True) # RNA polymerase concentration
22 c_b = Symbol('c_b', real=True) # BEWARE constant
23
24 # from Lai-Schaffer classic model
25
26 Shh = Symbol('Shh', real=True) # Shh quantity [0,30]
27 k_shh = Symbol('k_shh', real=True) # dissociation constant shh-ptc
    bindings [0.58,2.0]
28 k_ptc = Symbol('k_ptc', real=True) # half maximal concentration of ptc
    which inhibits smo signlaing
29 k_deg = Symbol('k_deg', real=True) # degradation constant for all gli
    related proteins
30 k_g3rc = Symbol('k_g3rc', real=True) # rate constant for the conversion
    to signal strengh
31 r_g3b = Symbol('r_g3b', real=True) # basal rate of Gli3 synthesis
32 K_g3rc = Symbol('K_g3rc', real=True) # sensitivity constant of the
    conversion to signal strengh
33 k_deg_p = Symbol('k_deg_p', real=True) # Degradation rate constant for Ptc
    [0.045,0.071]
34
35 gli = Symbol('gli', real=True)

```

```

36 gli3 = Symbol('gli3', real=True)
37 gli3R = Symbol('gli3R', real=True)
38 # Equation system
39
40 freg = (1-1/c+1/c*(1+a_gli*c*gli/k_gli+a_gli3*c*gli3/k_gli3R+r_gli3R*c*
      gli3R/k_gli3R)**3)/(1-1/c+1/c*(1+c*gli/k_gli+c*gli3/k_gli3R+c*gli3R/
      k_gli3R)**3)
41 beware = c_b/(1+k_RNAP/(RNAP*factor(expand(freg))))
42
43
44 print(latex(factor(expand(freg)),mode='plain'))
45 print(factor(expand(beware)))

```

A.2. Octave

Operador BEWARE en Octave.

```

1 function xdot = f (x,t)
2
3     Shh = 0.058*150 ; # Shh quantity [0,30]
4     k_shh = 0.58 ;# dissociation constant shh-ptc bindings [0.58,2.0]
5     k_ptc = 8.3*10**-2; # 1/2maximal concentration of ptc which inhibits
    smo signlaing
6     k_deg = 0.009 ;# degradation constant for all x(1) related proteins
7     k_g3rc = 0.012 ;# rate constant for the conversion to signal strenGh
8     r_g3b = 1.6*10**-1; # basal rate of x(2) synthesis
9     K_g3rc = 0.1 ; # sensitivity constant of the conversion to signal
    strenGh
10    k_deg_p = 0.09 ; # degradation rate constant for x(4) [0.045,0.071]
11    # -----
12    K1 = 8.3*10**-1;
13    K2 = 8.3*10**-1;
14    c = 1;
15    e = 0.5;
16    r = 0.2;
17    v_max = 2.4*10**-1;
18    r_bas = v_max/100;

```

```

19     v_maxp = 7.5*10**-1;
20     r_basp = v_maxp/100;
21     xdot(1) = v_max*((x(2)*K1+x(1)*K2)*(3***2*K1**2*K2**2+3*c*e*K1*K2*(
x(2)*K1+x(1)*K2+2*e*x(3)*K1*r)+c**2*(x(2)**2*K1**2+x(1)**2*K2**2+3*e*x
(1)*x(3)*K1*K2*r + 3***2*x(3)**2*K1**2*r**2 + x(2)*K1*(2*x(1)*K2 + 3*e*
x(3)*K1*r)))/(3*c*K1*K2*(x(2)*K1 + x(3)*K1 + x(1)*K2)**2 + c**2*(x(2)*
K1 + x(3)*K1 + x(1)*K2)**3 +K1**2*K2**2*(3*x(2)*K1 + 3*x(3)*K1 + (3*x(1)
+ K1)*K2)))+r_bas*((3*c*K1*K2*(x(2)*K1+ x(1)*K2 + x(3)*K1*r)**2 + c
**2*(x(2)*K1 + x(1)*K2 + x(3)*K1*r)**3 + K1**2*K2**2*(3*x(2)*K1 + 3*x(1)
*K2 + K1*(K2+ 3*x(3)*r)))/(3*c*K1*K2*(x(2)*K1 + x(3)*K1 + x(1)*K2)**2 +
c**2*(x(2)*K1 + x(3)*K1 + x(1)*K2)**3 + K1**2*K2**2*(3*x(2)*K1 + 3*x(3)
*K1 + (3*x(1) +K1)*K2)))-k_deg*x(1);
22     xdot(2) = r_g3b/x(4)-x(2)*(k_deg+(k_g3rc/(K_g3rc+((1+Shh/k_shh)/(1+
Shh/k_shh+x(4)/k_ptc))));
23     xdot(3) = x(2)*(k_g3rc/(K_g3rc+((1+Shh/k_shh)/(1+Shh/k_shh+x(4)/
k_ptc))))) -k_deg*x(3)
24     xdot(4) = v_maxp*((x(2)*K1+x(1)*K2)*(3***2*K1**2*K2**2+3*c*e*K1*K2
*(x(2)*K1+x(1)*K2+2*e*x(3)*K1*r)+c**2*(x(2)**2*K1**2+x(1)**2*K2**2+3*e*x
(1)*x(3)*K1*K2*r + 3***2*x(3)**2*K1**2*r**2 + x(2)*K1*(2*x(1)*K2 + 3*e*
x(3)*K1*r)))/(3*c*K1*K2*(x(2)*K1 + x(3)*K1 + x(1)*K2)**2 + c**2*(x(2)*
K1 + x(3)*K1 + x(1)*K2)**3 +K1**2*K2**2*(3*x(2)*K1 + 3*x(3)*K1 + (3*x(1)
+ K1)*K2)))+r_basp*((3*c*K1*K2*(x(2)*K1+ x(1)*K2 + x(3)*K1*r)**2 + c
**2*(x(2)*K1 + x(1)*K2 + x(3)*K1*r)**3 + K1**2*K2**2*(3*x(2)*K1 + 3*x(1)
*K2 + K1*(K2+ 3*x(3)*r)))/(3*c*K1*K2*(x(2)*K1 + x(3)*K1 + x(1)*K2)**2 +
c**2*(x(2)*K1 + x(3)*K1 + x(1)*K2)**3 + K1**2*K2**2*(3*x(2)*K1 + 3*x(3)
*K1 + (3*x(1) +K1)*K2)))-k_deg_p*x(4)
25 endfunction

```

Script para simular el sistema.

```

1 x = lode ("f", [1.76346322908;1.03692910138;2.81721868432;1.660269329], (t
= linspace (0, 2000, 20000)'));
2 plot(t,x)

```

A.3. XppAut

Script tipo *.ode* para simular los diagramas de bifurcación con el nuevo BEWARE.

```

1 F_reg_nt_coop=(1-1/c+1/c*(1+a_Gli*c*Gli/k_Gli+a_Gli3*c*Gli3/k_Gli3+r_Gli3R*
   c*Gli3R/k_Gli3R)**3)/(1-1/c+1/c*(1+c*Gli/k_Gli+c*Gli3/k_Gli3+c*Gli3R/
   k_Gli3R)**3)
2 Beware=c_b/(1+k_RNAP/(RNAP*F_reg_nt_coop))
3 Signal=(1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
4 Gli' =Beware-k_deg*Gli
5 Gli3' =r_g3b/Gli-Gli3*(k_deg+(k_g3rc/(H_g3rc+Signal)))
6 Gli3R' =Gli3*(k_g3rc/(H_g3rc+Signal))-k_deg*Gli3R
7 Ptc' =c_b1*Beware-k_deg_p*Ptc
8 par Shh=0.1,r_g3b=9.6,v_max=14.4,k_shh=1,k_ptc=0.083,k_deg=0.54,k_g3rc
   =0.72,H_g3rc=0.1,k_deg_p=5.4,K1=0.83,K2=0.83,c=1,e=0.5,r=0.2,r_bas
   =0.144,v_maxp=45.0,r_basp=0.45
9 init Gli=5,Gli3=0.06,Gli3R=0.7,Ptc=0.28
10 done
11
12
13 c = 1,a_Gli = 4.35,a_Gli3 = 4.35,r_Gli3R = 5*10**-5,k_Gli = 9*10**1,k_Gli3
   = 9*10**1,k_Gli3R = 9*10**1,k_RNAP = 1,RNAP = 1,c_b = 0.26,c_b1 = 3.15
14
15
16 Shh = 0.1,k_shh = 0.58,k_ptc = 8.3*10**-2,k_deg = 0.009,k_g3rc = 0.012,
   r_g3b = 1.6*10**-1,K_g3rc = 0.1,k_deg_p = 0.09

```

Script tipo *.ode* para simular los diagramas de bifurcación con el BEWARE clásico.

```

1 Basal=(3*c*K1*K2*(Gli3*K1+ Gli*K2 + Gli3R*K1*r)**2 + c**2*(Gli3*K1 + Gli*K2
   + Gli3R*K1*r)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli*K2 + K1*(K2+ 3*Gli3R*
   r)))/(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 +
   Gli3R*K1 + Gli*K2)**3 + K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli +K1
   )*K2))
2 Promoter=((Gli3*K1+Gli*K2)*(3*e**2*K1**2*K2**2+3*c*e*K1*K2*(Gli3*K1+Gli*K2
   +2*e*Gli3R*K1*r)+c**2*(Gli3**2*K1**2+Gli**2*K2**2+3*e*Gli*Gli3R*K1*K2*r
   + 3*e**2*Gli3R**2*K1**2*r**2 + Gli3*K1*(2*Gli*K2 + 3*e*Gli3R*K1*r))))
   /(3*c*K1*K2*(Gli3*K1 + Gli3R*K1 + Gli*K2)**2 + c**2*(Gli3*K1 + Gli3R*K1
   + Gli*K2)**3 +K1**2*K2**2*(3*Gli3*K1 + 3*Gli3R*K1 + (3*Gli + K1)*K2))
3 Signal=(1+(Shh/k_shh))/(1+(Shh/k_shh)+(Ptc/k_ptc))
4 Gli' =v_max*Promoter+r_bas*Basal-k_deg*Gli
5 Gli3' =r_g3b/Gli-Gli3*(k_deg+(k_g3rc/(H_g3rc+Signal)))
6 Gli3R' =Gli3*(k_g3rc/(H_g3rc+Signal))-k_deg*Gli3R

```

```
7 Ptc'=v_maxp*Promoter+r_basp*Basal-k_deg_p*Ptc
8 par Shh=0.1,r_g3b=9.6,v_max=14.4,k_shh=1,k_ptc=0.083,k_deg=0.54,k_g3rc
   =0.72,H_g3rc=0.1,k_deg_p=5.4,K1=0.83,K2=0.83,c=1,e=0.5,r=0.2,r_bas
   =0.144,v_maxp=45.0,r_basp=0.45
9 init Gli=5,Gli3=0.06,Gli3R=0.7,Ptc=0.28
10 done
```