Aprendizaje de redes regulatorias bacterianas a partir de algoritmos genéticos

Director: Igor Zwir

Co-Director: Oscar Harari

Licenciatura en Ciencias de la Computación

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Alumno: **Marcelo Santos**
marcesantos@gmail.com

Abstract

Gene expression is determined by protein-protein interactions among several regulatory proteins and by RNA polymerase(s), and protein-DNA interactions of these molecules with DNA sequences in the promoters of regulated genes [2]. These interactions define complex genetic networks, whose designs have motivated researchers to use mathematical and computational tools for the construction and posterior analysis of a proposed network diagram. This complexity is increased by the number of genes considered in the networks and the amount of noise that exists in available data.

The development of computational models of genetic networks has a tremendous importance since it allows, among other applications, to obtain dynamic models of living systems to facilitate discovery and characterization of potential therapeutic targets.

The network topology is not the only determinant element in the dynamics of gene expression, but also are crucial the parameters of the model. Indeed, in different contexts of cellular biology it has been proposed that molecular interaction nets inside the cell should have an intrinsic robustness regarding the parameter variations that define them. Because of the extent of the search space of these parameters it is not viable to apply exhaustive techniques to study the possible configurations.

Nowadays, there is a method that deals with those problems using an adaptative approach. In order to carry out this task this method uses a random search process to get initial conditions and parameters of the model. As a result of this approach this method has a low performance shown by the low ratio of valid solutions obtained and the total number of evaluations of models.

On the other hand, there is a set of computational models that allows the resolution of certain problems based on evolutive process models which are known as evolutive algorithms [1]. Among this set we can find genetic algorithms which are a metaheuristic technique for the solution of optimization problems. They are based on an analogy of species evolution theory, using this idea in order to search for one or more optimal solutions among a set of possible ones.

In this work we propose a method based on these algorithms with the aim of optimizing the capability to obtain initial conditions and parameters for genetic networks, focusing on their biological validity and the robustness of the obtained architecture.

To test this method we used a regulatory networks which is formed by a system of two components PhoP/PhoQ and PmrA/PmrB, which govern virulence and the adaptation to low Mg^{2+} and high Fe^{3+} environments

respectively, in the enteric bacteria *Salmonella enterica* serovar Typhimurium[8], [14]. The study of the PhoP regulon constitutes a special challenge due to the multiplicity of PhoP-controlled targets and the connectivity of the PhoP/PhoQ system with other two-component systems, such as PmrA/PmrB, transcriptional regulators, and alternative RNA polymerase sigma factors.

The results obtained by the method described in this work showed a high correlation to the values collected in biological experimentations, validating the biological significance of our results. In addition, our method uses a learning approach which does not need extra information to reach its goal; and it obtained better numerical scores and better proportion of valid solutions than the random search approach. Finally, these architectures showed a great robustness regarding their parameters configuration.

Resumen

La expresión genética está determinada por interacciones de tipo proteína-proteína entre diversas proteínas regulatorias y la(s) ARN polimerasa(s), e interacciones de estas con secuencias de ADN de los sitios promotores de los genes regulados [2]. Estas interacciones definen redes genéticas complejas cuyos diseños han motivado a los investigadores a utilizar herramientas matemáticas y computacionales para la construcción y posterior análisis de los diagramas de interacciones. Esta complejidad se ve incrementada por la cantidad de genes considerados en las redes y por el ruido existente en los datos disponibles.

El desarrollo de modelos computacionales de redes genéticas posee una gran importancia dado que posibilita, entre otras aplicaciones, obtener modelos dinámicos de sistemas vivientes que facilita el descubrimiento y caracterización de blancos terapéuticos potenciales.

No sólo la topología de una red es determinante de la dinámica en la expresión de los genes sino también los parámetros del modelo, problema de gran dimensionalidad para ser abordado con técnicas exhaustivas. A su vez, en distintos contextos dentro de la biología celular se plantea que las redes de interacción molecular dentro de las células deberían tener una robustez intrínseca a las variaciones de los parámetros que la definen.

Actualmente existe un método que permite manejar estos problemas a través de un enfoque adaptativo. Para llevar a cabo esta tarea, este método utiliza una aproximación por búsqueda aleatoria para obtener las condiciones iniciales y parámetros del modelo, lo cual lo hace poco performante ya que la proporción de ejecuciones de una red en relación a las soluciones válidas es muy baja.

Por otro lado, existe un conjunto de modelos computacionales que permiten la resolución de problemas en base a modelos de proceso evolutivo y que se los conoce genéricamente con el nombre de algoritmos evolutivos [1]. Dentro de ellos se encuentran los algoritmos genéticos, que son una técnica metaheurística para la solución de problemas de optimización y que se basan en una analogía de la teoría biológica de la evolución de las especies, utilizando esta idea para buscar una o más soluciones óptimas entre un conjunto de posibles soluciones.

En el presente trabajo proponemos un método basado en dichos algoritmos con el objetivo de optimizar la obtención de condiciones iniciales para redes genéticas y de esa manera lograr soluciones válidas al problema planteado, en lo referente a validez biológica y robustez de las arquitecturas que se obtengan.

Para probar este método se utilizó una red regulatoria que está formada por los sistemas de dos componentes PhoP/PhoQ y PmrA/PmrB, los cuales gobiernan la virulencia y la adaptación a medios de bajo Mg^{2+} y alto Fe^{3+}, respectivamente, en la bacteria Salmonella enterica serovar Typhimurium[8] [14]. El estudio del regulón PhoP constituye un desafío especial debido a la multiplicidad de blancos controlados por PhoP, y a la

conectividad del sistema PhoP/PhoQ con otros sistemas de dos componentes, tales como PmrA/PmrB, reguladores transcripcionales y factores sigma alternativos de RNA polimerasa.

Los resultados arrojados por nuestro método mostraron una altísima correlación entre estos y los valores obtenidos en experimentaciones biológicas, lo que demuestra que el método tiene una gran validez biológica. A esto se le agrega que el mecanismo de aprendizaje utilizado no necesita información extra para lograr su objetivo. Por otro lado, el método obtuvo una mayor proporción de resultados en relación a los obtenidos por el método basado en búsqueda aleatoria, con mejores puntajes (*scores*) numéricos. Finalmente, estas arquitecturas mostraron una gran robustez en cuanto a la configuración de los parámetros definidos.

Agradecimientos

Antes de comenzar con el desarrollo de esta tesis de Licenciatura quisiera realizar algunos agradecimientos, ya que me hubiera sido imposible terminar este trabajo sin el apoyo, tanto científico como humano, de varias personas.

En primer lugar quisiera agradecerle a Igor por aceptar dirigir esta tesis y proveerme las bases biológicas y computacionales fundamentales para este trabajo, además de aportarme su visión y propia experiencia en esta área para permitirme continuar ante lo que me parecían problemas difíciles de resolver.

También quisiera agradecerle a Oscar por colaborar en las correcciones de los documentos que generaba, además del estímulo que me brindó para terminar esta tesis.

A Patricio le agradezco su apoyo en el inicio de este trabajo para comprender las características del problema que intentaba optimizar, así como del material que me fue indispensable para comenzar con el mismo.

A mi familia por su apoyo y preocupación constantes, sobre todo en los momentos en los que mi salud hizo pasar a esta tesis a un segundo plano.

A mis compañeros de estudio a lo largo de la carrera, en especial a Fernando por su capacidad de motivación y optimismo, los cuales siempre me fueron contagiosos.

Índice General

Índice de Figuras

Índice de Cuadros

Capítulo 1

Bases Biológicas

1. Introducción

Todos los seres vivos están constituidos por células, que son las unidades básicas organizacionales cuya maquinaria permite mantener los procesos vitales y asegurar la reproducción de individuos. Si bien, la variabilidad entre distintas familias y especies de seres vivos es sorprendente, lo es aún más que los mecanismos que utilizan éstas para el mantenimiento de la vida es muy similar entre ellas, así como el código utilizado para mantener la información de su maquinaria.

En el presente capítulo daremos una introducción a los conceptos principales de la estructura y dinámica biológica y molecular de las células y los genes. Luego, hacia el final del capítulo explicaremos un caso particular de regulación de la expresión genética como lo son los sistemas de dos componente PhoP/PhoQ – PmrA/PmrB.

2. Bases Biológicas

2.1. Las células

Las células han evolucionado en los últimos 3.500 millones de años, presumiblemente a partir agregaciones moleculares. El mundo de los seres vivos está constituido por dos tipos de células; *procariotas* y *eucariotas*. Ambos tipos se diferencian principalmente en la complejidad de su maquinaria biológica. Así por ejemplo, éstas últimas poseen una estructura denominada núcleo en donde guardan su información genética, la cual está dividida en ciertas estructuras llamadas cromosomas, mientras que las células procariotas no poseen núcleo y toda su información genética está localizada en un único genoma circular.

Las células y sus componentes están protegidos del medio exterior por una membrana llamada *membrana plasmática*, la cual posee una estructura química en forma de doble capa lipídica. Los lípidos son un conjunto de biomoléculas insolubles en agua, compuestas principalmente por carbono e hidrógeno y en menor medida por oxígeno y otros componentes. A su vez en esta membrana existen distintas proteínas, que son cadenas de aminoácidos con distintas funciones biológicas (ver sección 2.2.2). Las proteínas que se encuentran asociadas a la membrana celular pueden cumplir las siguientes funciones:

- Constituir canales que permitan la entrada y salida de moléculas.
- Dar estabilidad a partir de uniones entre ellas.
- Actuar como receptores para señales que activen o inhiban procesos celulares.

- Actuar en reacciones químicas (enzimáticas) a partir de señales externas.

En la figura 1.1 se muestra un esquema de dicha membrana.

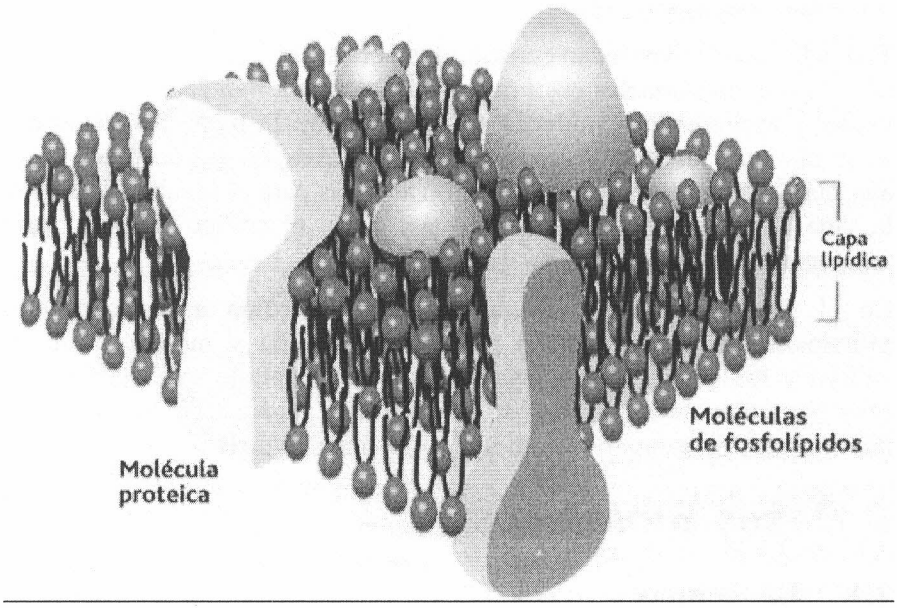


Figura 1.1. Membrana plasmática. Obsérvese la composición en forma de doble capa lipídica en donde se ven también moléculas de proteínas, las cuales tienen diversas funciones.

2.2. El ADN

2.2.1. Estructura del ADN

La información de la maquinaria celular está codificada dentro de la célula en forma de ADN, la cual tiene una estructura de doble cadena, es decir, dos largos polímeros paralelos no ramificados donde cada elemento (o monómero) recibe el nombre de nucleótido. Estos están formados por (Figura 1.2):

- Un grupo fosfato
- Un residuo de azúcar (desoxirribosa)
- Una base nitrogenada (adenosina, guanina, citosina o timina)

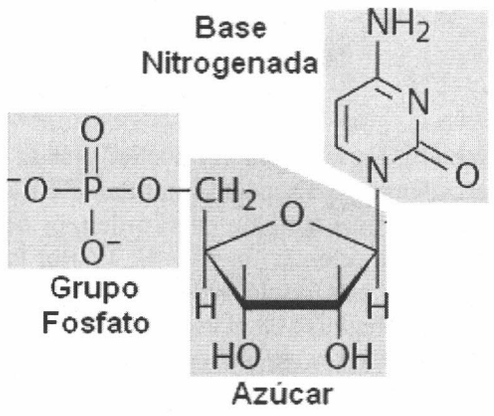


Figura 1.2. Estructura de un nucleótido.

Cada una de las cadenas de nucleótidos se forma por uniones entre el grupo fosfato (el cual se encuentra unido al átomo de carbono en posición 5') de uno de sus componentes y el átomo de carbono que se encuentra en la posición 3' de la desoxirribosa del elemento contiguo, como se muestra en la figura 1.3. Las posiciones en las que se producen la uniones son importantes ya que en uno de los extremos de la cadena habrá un grupo fosfato que no se une a ninguna desoxirribosa, mientras que en el otro, el carbono en posición 3' del azúcar estará libre. Esta disposición permite tener un orden en el que se leen las distintas bases nitrogenadas, 3' – 5' o 5' – 3'.

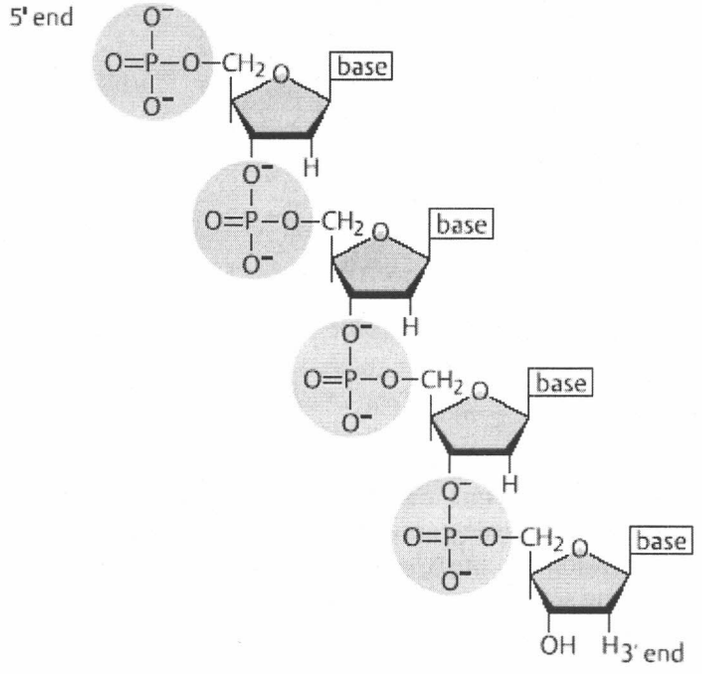


Figura 1.3. Estructura de una cadena de nucleótidos.

La unión de la doble cadena se basa en la complementariedad entre las bases nitrogenadas, así la adenosina de una cadena formará uniones con la timina de la cadena contigua y la citosina lo hará con la guanina (A-T

y C-G respectivamente). En la figura 1.4 puede verse la organización del ADN como doble cadena. Otro aspecto importante a destacar es que las cadenas son complementarias también en el orden, es decir mientras una de las cadenas tendrá una orientación 3' – 5', la otra lo hará en sentido inverso 5' – 3'. Además esta doble cadena se presenta con una forma helicoidal como puede apreciarse en el esquema de la figura 1.4.[20]

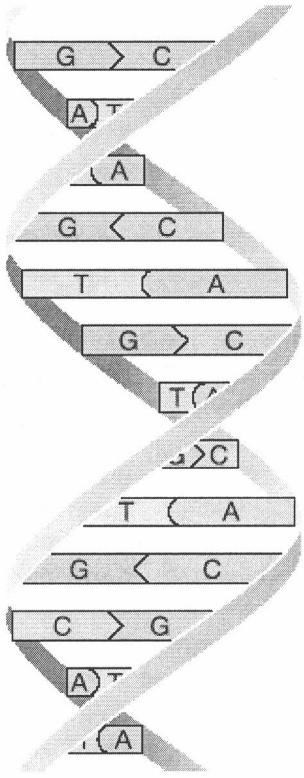


Figura 1.4. Estructura de la doble cadena de ADN.

2.2.2. La información genética

La información que contiene el ADN permite a la célula construir moléculas de proteínas que cumplen en la célula variadas funciones, como ser, catalizadoras de reacciones químicas (*enzimas*), estimular o inhibir la expresión de otros genes, mantener estructuras celulares y tisulares, traducir señales, facilitar el movimiento, etc. Las proteínas son cadenas de aminoácidos, los cuales tienen una estructura que les permiten unirse unos a otros formando cadenas o *polipéptidos*. En la naturaleza existen 20 aminoácidos distintos, lo que permite que haya una gran variabilidad de polipéptidos que se pueden producir. Estos a su vez, según sea la secuencia de aminoácidos que la forman, pueden plegarse formando estructuras tridimensionales y/o también unirse a otros polipéptidos u otras moléculas que le permitan cumplir con su función. Así por ejemplo, la hemoglobina es una proteína formada por cuatro cadenas peptídicas y un grupo central (*grupo hem*), el cual es el que se

une al oxígeno y permite transportarlo por el torrente sanguíneo hacia los distintos tejidos del cuerpo.

El código genético está formado por la secuencia de las bases nitrogenadas que se encuentran en la cadena de ADN, lo cual determina que se trate de un alfabeto de 4 letras. Tres bases consecutivas forman un *codón*, el cual codifica para un único aminoácido[1].

Para transformar la información contenida en el ADN a proteínas se necesitan dos grandes pasos, la *trascripción* y la *traducción*. La primera de ellas consiste en transformar las cadenas de nucleótidos del ADN en una estructura intermedia, llamada *ARN mensajero* (ARN_m), para luego traducir este último a una cadena polipeptídica.

El ARN (Ácido Ribonucleico) es una cadena simple de nucleótidos con una estructura similar al ADN, con la diferencia de que el azúcar que posee cada componente es la ribosa, en lugar de la desoxirribosa, y no puede tener timina como base nitrogenada, sino que utiliza el uracilo, el cual es también complementario a la adenosina.

La trascripción del ADN al ARN_m se realiza también por complementariedad de bases. Básicamente lo que sucede es que la doble cadena helicoidal del ADN se abre y la cadena con dirección 5' – 3' es utilizada como molde para la trascripción (Figuras 1.6), de tal forma que cuando se lea una adenosina en la cadena de ADN se agrega un uracilo en el ARN, cuando se encuentra una timina, se agrega una adenosina y así siguiendo.

En los organismos procariotas estas acciones son llevadas a cabo por una enzima, la *ARN Polimerasa*, que para poder cumplir su tarea necesita adherirse al ADN en ciertos sitios específicos, para así poder abrir la doble cadena y comenzar a producir el ARN_m. Esta zona del ADN en la que se adhiere la ARN Polimerasa se lo llama *promotor* (Figura 1.5) y puede estar alojado en distintas regiones por delante de los genes a transcribir (*upstream* o corriente arriba). Llamativamente, es encontró que en muchos de los organismos estudiados existen 2 zonas en las que se une la ARN Polimerasa, las cuales están en las posiciones 35 y 10 corriente arriba y poseen una secuencia de bases similares (*secuencias de consenso*).

Un aspecto a destacar es que una molécula de ADN puede permitir producir muchos ARN_m, los cuales tienen un tiempo de vida determinado. De esta manera el ADN actúa como una memoria de sólo lectura para la célula.

---

[1] Como se puede ver la cantidad de codones posibles (64) es mayor a la cantidad de aminoácidos que existen (20), sin embargo, algunos aminoácidos tienen más de una codificación posible y existen ciertos codones que son utilizados como señales y no codifican para ningún aminoácido.

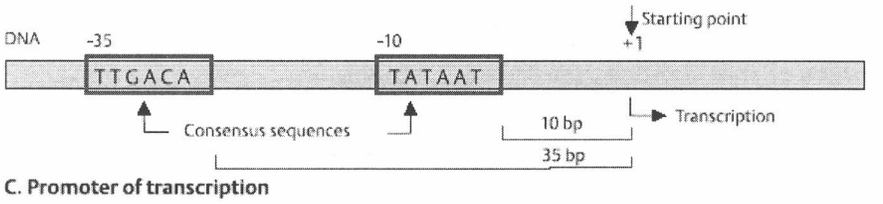


Figura 1.5. Esquema de un promotor para la ARN Polimerasa.

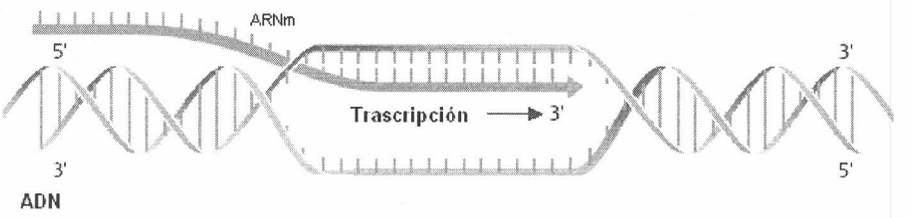


Figura 1.6. Esquema del mecanismo de trascripción.

Para realizar la traducción del ARN_m a una cadena polipeptídica, es necesaria la participación de otro tipo de ARN, el ARN de transferencia (ARN_t). Existe distintos tipos de ARN_t, y cada uno de ellos puede unirse a un aminoácido determinado, por sus extremos, con lo que cada de éstos tendrá un ARN_t específico. Además los ARN_t poseen una secuencia específica de tres nucleótidos, llamada *anticodón*, la cual le permite reconocer un codón o subgrupo de codones del ARN_m por emparejamiento de bases. En la figura 1.7 se esquematiza la estructura de un ARN_t.

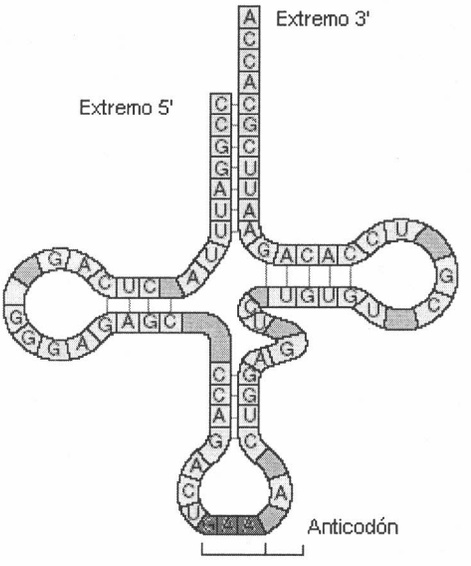


Figura 1.7. Esquema del ARNt.

Para la síntesis de proteínas, el ARN_m va siendo leído desde el extremo inicial de la cadena, el cual posee un codón AUG, y luego para cada codón leído se utilizan moléculas de ARN_t cargadas con sus aminoácidos respectivos que se une al ARN_m por emparejamiento de sus anticodones

con cada uno de los codones (Figura 1.8). Después, los aminoácidos se van uniendo de forma que la proteína naciente va creciendo y cada ARN_t, liberado de su carga, se separa de la estructura.[20]

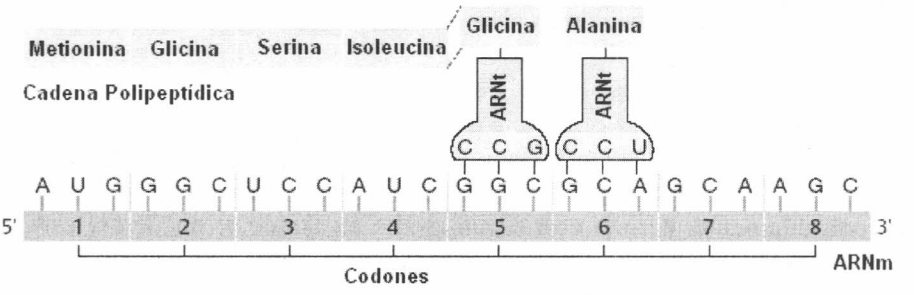


Figura 1.8. Esquema del mecanismo de traducción.

Las moléculas de ADN son muy largas y contienen la especificación de miles de proteínas. Por tanto, fragmentos de esta secuencia completa de ADN se transcriben en diferentes moléculas de ARN_m. Un *gen* se define como un fragmento de la secuencia de ADN que corresponde a una sola proteína (o a una molécula de ARN catalítica o estructural, para los genes que son trascriptos a ARN pero no traducidos luego a proteínas).

2.2.3. Regulación de la actividad genética

En todas las células, la expresión de determinados genes está regulada: en lugar de sintetizar el catálogo completo de posibles proteínas en todo momento, la célula ajusta la velocidad de trascripción y de traducción de diferentes genes de forma independiente y de acuerdo con sus necesidades. La regulación de la expresión de los genes es una función básica tanto en los organismos eucariotas como procariotas. Estos últimos dependen completamente en su habilidad para adaptarse rápidamente a los cambios en las condiciones externas, ya que sustancias necesarias que no se encuentren disponibles deberán ser sintetizadas por ellos mismos.

Un aspecto importante de las cadenas de ADN es que no todas los nucleótidos forman parte de genes, sino que existen secuencias enteras que se encuentran entre conjuntos de genes, a las que se las denominan espacios intergénicos. Muchas de estas cadenas no tienen una funcionalidad conocida, pero se sabe que algunas constituyen sitios de unión de proteínas que pueden estimular o inhibir la transcripción de los genes cercanos.

El control de la expresión genética ocurre a diferentes niveles. Algunas proteínas pueden actuar como represores de la actividad de la ARN polimerasa o como activadores de ésta. En general estos mecanismos se dan por una competencia por la unión al promotor con dicha enzima. El control de la expresión genética en los procariotas es, usualmente, facilitada por genes relacionados funcionalmente que están contiguos en la cadena de ADN y pueden ser regulados en conjunto (*operón*).

Frente a la presencia de ciertas sustancias en el medio, se puede inducir la síntesis de ciertas enzimas para que los organismos procariotas puedan utilizarlas. Por ejemplo, la bacteria Escherichia coli posee tres enzimas encargadas del catabolismo de la lactosa, cuya producción puede estimularse frente a la presencia de lactosa en el medio. Por otro lado, un gen puede inhibir su trascripción frente al cambio en el medio. Por ejemplo, el triptofano es un aminoácido que E. Coli produce constantemente, pero si está en un medio en donde se encuentra éste, la actividad enzimática para la biosíntesis del triptofano decrece rápidamente, por inhibición de la producción de 5 genes que forman un operón.

3. Sistemas de dos componentes

Una pregunta fundamental en biología es cómo un organismo integra múltiples señales para generar una respuesta celular apropiada. Para cada conjunto de señales, existe un cierto número de genes (no siempre conocido en su totalidad) que codifican las distintas especies proteicas encargadas de capturar el estímulo (*input*), realizar la transducción de señales necesaria para activar la maquinaria celular de respuesta, y finalmente generar la respuesta adecuada (*output*), generalmente resumida en una variación en la expresión de nuevos genes. Estos conjuntos de genes determinan verdaderos circuitos biológicos de control celular.

En los procariotas, una importante cantidad de funciones celulares es controlada por sistemas regulatorios de dos componentes. Circuitos dedicados traducen e interpretan señales específicas tales como pH, temperatura, osmolaridad, luz, nutrientes, iones, feromonas, y toxinas para regular un amplio rango de procesos que incluyen movilidad, virulencia, metabolismo, ciclo celular, interruptores (*switches*) de desarrollo, resistencia a antibióticos y respuesta a stress [8].

Estos sistemas de dos componentes constituyen el mecanismo principal de transducción de señales que permiten a las bacterias modificar su comportamiento celular en respuesta a estímulos del ambiente. Los dos componentes consisten en:

- Una proteína sensora con actividad histidin-kinasa que responde a señales específicas del medio fosforilando a su proteína asociada en el sistema.
- Una segunda proteína que, una vez activada, es decir, fosforilada, por la primera, constituye un factor de trascripción capaz de asociarse al ADN de ciertos genes regulados por este sistema, estimulando la transcripción de los mismos.

La asociación del primer componente con un ligando específico (estímulo) modifica tres actividades enzimáticas de esta proteína sensora:

1. Actividad autokinasa: Capacidad de autofosforilarse en presencia de ATP.

2. Actividad kinasa: Transferencia de este grupo fosforil al segundo componente del sistema.
3. Actividad fosfatasa: Defosforilación del segundo componente disminuyendo su actividad regulatoria de la trascripción.

De esta manera, la fosforilación o defosforilación de estas proteínas puede modificar la expresión de ciertos genes de la bacteria, y una señal específica del medio se traduce en una señal de respuesta de la bacteria en la forma de una modificación en su perfil de expresión.

Dos modelos biológicos ampliamente estudiados son los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ, tanto en su funcionamiento individual como en su interacción [14]. El caso de la acción coordinada de estos sistemas reviste especial interés dado que presenta las características de una verdadera red genética, conservando una relativa simpleza estructural.

3.1. Los sistemas de dos componentes PhoP/PhoQ y PmrA/PmrB

El sistema de dos componentes PmrA/PmrB (regulador de respuesta/sensor, respectivamente) presente en Salmonella enterica serovar Typhimirium, organismo tomado como modelo para la implementación del método que presentaremos en el presente trabajo, es necesario para la resistencia inducida al antibiótico polimixina B, la resistencia a la muerte mediada por Fe^{3+}, el crecimiento en suelo, la virulencia en ratones, y la infección de macrófagos de pollo. Responde independientemente a dos señales, alto nivel de Fe^{3+} extracelular, censado por PmrB, y bajo nivel de Mg^{2+}, censado por la proteína PhoQ.

El sistema PhoP/PhoQ (regulador de respuesta/sensor, respectivamente), presente en el mismo organismo, constituye un regulador maestro (regulando aproximadamente la trascripción del 2% del genoma de la bacteria) que gobierna la adaptación a medios de bajo Mg^{2+} y la virulencia en ratones así como también otras funciones biológicas. El estímulo activador para este sistema es el bajo nivel de Mg^{2+} extracelular. Entre los genes activados por PhoP se encuentra el gen pmrD, el cual resulta de especial interés porque presenta un sitio de vinculación (*binding site*) para PmrA, y su producto, la proteína PmrD, puede asociarse a la proteína PmrA. Ambos sistemas de dos componentes muestran una coordinación de sus funciones in vivo, aunque los mecanismos exactos de interacción son aún desconocidos. Un esquema de su funcionamiento puede apreciarse en la Figura 1.9.

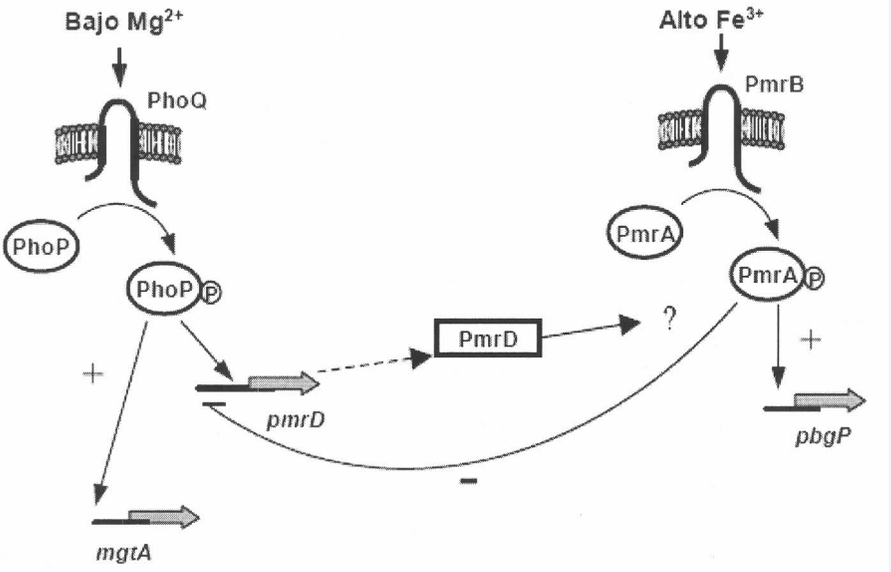


Figura 1.9: Esquema genérico de funcionamiento de los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ.

4. Comentarios finales

En este capítulo hemos dado una introducción a los distintos conceptos biológicos que manejaremos a lo largo del presente trabajo. Entre ellos se destacaron las características de expresión genética en base a procesos de trascripción del ADN en ARN_m y traducción a proteínas a partir de éstos. Por otro lado se explicaron los mecanismos de regulación de la actividad genética, poniendo especial atención en dos sistemas de dos componentes hallado en la bacteria Salmonella enterica serovar Typhimirium, como son los sistemas PhoP/PhoQ y PmrA/PmrB. Estos sistemas serán utilizado en futuros capítulos para mostrar el comportamiento del método que nos proponemos presentar en este trabajo.

Capítulo 2

Bases Computacionales

1. Introducción

Es posible afirmar que el gran desafío de la era postgenómica puede ser sintetizado respondiendo preguntas del tipo *cuándo*, *dónde*, y *por cuánto tiempo* [3] un gen esta prendido o apagado. En la actualidad existe un alto grado de incertidumbre en cuanto a los mecanismos precisos que determinan las respuestas a estas preguntas. Es debido a esto que se requiere la ayuda de computadoras para realizar simulaciones que guíen a los investigadores en estos aspectos.

En este capítulo explicaremos las bases computacionales para el modelado de redes genéticas (sección 2), en la que se describirán las generalidades y distintas aproximaciones posibles, dejando para el final de esta sección la explicación de la opción elegida para modelar los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ de *Salmonella enterica*.

En la sección 3 se hará una introducción a los algoritmos genéticos, como técnica para la optimización de diversos problemas, entre los que se encuentra el que intentamos resolver en el presente trabajo. Finalmente, en la sección 4, plantearemos las características de los problemas multiobjetivo, para tener una mejor comprensión de sus características cuando planteemos nuestro método en el capítulo 4.

2. Modelado de redes genéticas

2.1. Introducción

Cada vez que resolvemos un problema correspondiente al mundo real, como es el caso de los problemas biológicos, debemos comprender que, en realidad, solamente, estamos encontrando la solución de un modelo determinado del problema. Es decir, todos los modelos son simplificaciones del mundo real, de otro modo serían tan complejos e intrincados como lo es el escenario natural. Uno de los problemas con los que se tiene que lidiar cuando se definen modelos es que cada uno de ellos debe hacer una serie de asunciones respecto del problema real, lo cual implica decidir que elementos son los relevantes y cuales pueden eliminarse del mismo. En consecuencia, el error más frecuente que existe cuando tratamos con modelos es olvidar dichas asunciones. En este sentido, un problema puede tener varias soluciones correctas posibles, dependientes de cómo fue construido el modelo del mismo.

El desarrollo de modelos matemáticos de sistemas reales es un tópico central en diferentes disciplinas como la ingeniería y las ciencias, dichos

modelos sirven para resolver problemas reales a través de simulaciones, análisis del comportamiento de un sistema, diseño de nuevos procesos para controlar sistemas, predicciones, etc.

A continuación veremos algunas propuestas existentes para el modelado de redes genéticas.

2.2. Aproximaciones al modelado de redes genéticas

Como se vio en el capítulo anterior, uno de los objetivos de los estudios biológicos de este último tiempo es el de comprender cómo es la dinámica de activación de genes, lo que está llevando a determinar una cantidad de circuitos genéticos cuya complejidad requiere de nuevos métodos para poder interpretarlos y deducir ciertos aspectos que experimentalmente aún no es posible conocer.

El desarrollo de modelos computacionales de redes genéticas posee una gran importancia, dado que posibilita, entre otras aplicaciones, desarrollar modelos dinámicos de sistemas vivientes que, mediante la aplicación de distintas técnicas de data mining, permiten acelerar el proceso de drug discovery, facilitando el descubrimiento y caracterización de blancos terapéuticos potenciales.

Actualmente existe una gran cantidad de formas de modelar redes genéticas. En este modelado, se deben tener en cuenta, entre otras, las siguientes preguntas:

- ¿Qué variables deben ser consideradas en el modelo y cuales podrían dejarse de lado?
- ¿Cómo va cambiando la red de estado a lo largo del tiempo? ¿Es decir, cómo son las propiedades de la dinámica de la red?
- ¿Cuál es el estado de equilibrio del modelo?

Con respecto a la primera pregunta, habíamos dicho que un modelo es una simplificación de la realidad, con lo que es necesario definir que elementos no van a ser considerados en el mismo. Por ejemplo, en muchos casos no es tenido en cuenta la disponibilidad de energía (ATP – Adenosín Trifosfato) o cofactores para la actividad enzimática descripta en las redes, pero si las concentraciones de cada una de las especies que componen la red.

A grandes rasgos, podríamos decir que un modelo de redes genéticas tiene una topología o arquitectura, es decir, una descripción de cómo están relacionados los elementos y una dinámica que describe estas interacciones. El primero de estos aspectos, puede ser visto como un grafo en el cual las distintas especies moleculares de la red son representadas por los nodos, y las interacciones entre estas especies son representadas por los ejes. A su vez, estos pueden tener valores que representan la concentración de la especie representada, para los primeros y valores de interacción entre dos especies, para los ejes.

Según el tipo de valor que se asigne y las características de la dinámica de las redes, tendremos distintas aproximaciones, de las cuales nombraremos las siguientes:

- Modelos de redes booleanas: En este caso cada gen está totalmente expresado o no expresado del todo, es decir cada nodo posee 1 o 0 como valor y las interacciones entre especies están representadas por funciones booleanas, es decir, los valores para cada especie en un estado en particular van a estar determinados por el resultado de la aplicación de estas funciones sobre los valores de las especies del estado anterior.

  Estos modelos son utilizados para obtener una primera representación del problema a tratar sobre sistemas complejos con muchos elementos.

- Modelos de lógica cinética: En este caso los valores de las especies son representados con valores discretos, como ser "No expresado", "Expresado en un nivel bajo", "Expresado en un nivel medio", "Totalmente expresado", por ejemplo. Esto da una mayor granularidad que las redes booleanas. En este modelo, en lugar de que todos los genes cambien de estado al mismo tiempo, se permite que las especies cambien de estado según distintas tasas.

- Modelos de lógica continua: En este modelo, los valores de los nodos también son discretos pero la transición entre estados está definida por ecuaciones diferenciales lineales con coeficientes constantes.

- Modelos de ecuaciones diferenciales: Estos modelos permiten una generalización para casi cualquier reacción química. En este caso los valores de los nodos son variables continuas que representan las concentraciones de las distintas especies y las transiciones entre estados son consideradas como continuas y determinísticas.

- Modelos estocásticos: En este caso las reacciones ocurren con cierta probabilidad, lo que presenta un modelo probabilístico a diferencia de los anteriores que son determinísticos.

La calidad de la red obtenida no está determinada solamente por el modelo elegido, sino también por el diseño de un método de inferencia (estrategia de aprendizaje) adecuado que permita estimar los parámetros de la red. Una de las mayores dificultades en este sentido es causada por la dimensión del problema; las distintas estrategias para superar esta cuestión se basan en la reducción del número de elementos modelados (agrupamiento por ejemplo), el incremento de la cantidad de muestras (microarray) [19] o la simplificación de la complejidad del modelo (conectividad limitada) [24].

2.3. Enfoques actuales para el modelado computacional

2.3.1. Sistemas de ecuaciones diferenciales ordinarias con valores reales

Un caso particularmente interesante de modelado de redes genéticas es aquel en el que se utilizan valores continuos para determinar los niveles

de expresión de los compuestos y de las relaciones entre los mismos. Esto permite capturar aspectos biológicos observados experimentalmente, que son sacrificados en pro de la simplicidad en los modelos booleanos como el propuesto en [7].

Como se explicó anteriormente, una posible aproximación a estos modelos continuos es la basada en sistemas de ecuaciones diferenciales ordinarias (ODEs, del inglés Ordinary Differential Equation) no lineales, en la cual se proponen sistemas de ecuaciones que permiten calcular la variación en la concentraciones de las especies, ARN o proteína (valor de un nodo), en función del tiempo a partir de los valores de los nodos conectados con él. De esta manera, pueden realizarse simulaciones temporales del sistema, permitiéndose observar aspectos dinámicos del mismo, esenciales en los sistemas de control. Mediante estas simulaciones, es posible extraer propiedades emergentes a nivel sistémico, tales como la robustez de la red.

La bibliografía existente sobre este tema puede dividirse según el uso de modelos "estáticos" o "dinámicos" de ODEs. En los modelos de ODEs estáticos, como el descripto en [2], se propone que el sistema biológico alcanza un estado de equilibrio (es decir, en donde el diferencial de las concentraciones de los productos de sus genes en función del tiempo es igual a cero), en el cual se establecen las ecuaciones específicas del modelo, determinándose consiguientemente la red asociada. En los modelos dinámicos (como el propuesto en [23]) no se considera necesariamente este equilibrio, con lo que pueden observarse variaciones en los valores de expresión de los genes (valores de los nodos) a lo largo del tiempo. Esta última característica es la que permite realizar simulaciones temporales del comportamiento del sistema, siendo este hecho de especial interés al estudiar sistemas biológicos de regulación tales como PmrA/PmrB y PhoP/PhoQ, en los que puede observarse experimentalmente una respuesta dinámica a variaciones en los estímulos recibidos (variaciones en los valores de los nodos de entrada).

La noción de que las complejas redes de interacción molecular dentro de las células deberían ser robustas a variaciones de los parámetros que las definen aparece en distintos contextos dentro de la biología celular. En este sentido, suele sugerirse que esta robustez debería ser intrínseca a la arquitectura de la red, es decir que esta debería preservar sus características funcionales al producirse perturbaciones en uno o más de sus parámetros.

En las secciones siguientes se analizan distintos enfoques que permiten modelar redes genéticas en base a sistemas de ecuaciones diferenciales. En el primero, se utiliza un sistema de ecuaciones estático, en el sentido de que sólo se trabaja sobre aspectos matemáticos de las mismas (sin modificar sus parámetros) no realizándose pruebas que permitan determinar el comportamiento del sistema obtenido. En el segundo, se plantea un sistema dinámico que permite realizar simulaciones sistemáticas para determinar no sólo el comportamiento del sistema bajo distintas condiciones, sino también los valores de los parámetros de las mismas.

2.3.2. *Modelos estáticos*

En modelos basados en ecuaciones diferenciales como el propuesto en [2] para el sistema de dos componentes EnvZ/OmpR, se sugiere que la robustez de la red se evidencia en que el resultado de la misma no se ve afectado significativamente al variar los valores de las concentraciones de las especies de entrada. En este contexto, un conjunto de nodos es distinguido como "nodos de entrada" en el sentido de que sus miembros representan la interacción entre el entorno y la red genética (funcionando como receptores de señales); otro conjunto es distinguido como "nodos de salida" significando que el valor de expresión de sus miembros representa la respuesta a las señales (funcionando como el efecto de la transducción de señales). De esta manera, se sugiere una robustez con respecto a los valores iniciales de los nodos de entrada del sistema. Este enfoque, sin embargo, adolece de los siguientes defectos:

- Los parámetros de las ecuaciones diferenciales que definen este tipo de modelos no son generalmente conocidos, por lo que deben ser estimados en forma relativamente arbitraria.
- Una vez obtenido el modelo, resulta complejo, cuando no imposible, validar estos valores experimentalmente.
- El modelo es estático, dado que se asume que el sistema se encuentra en un estado estable en el cual no hay variaciones temporales en las concentraciones, por ejemplo, si bien EnvZ puede estar tanto fosforilado como defosforilado, se asume que la suma de las concentraciones de ambas especies permanece constante, no existiendo creación (síntesis) ni destrucción (decaimiento) de EnvZ. Esto impide detectar aspectos dinámicos del sistema, inherentes a todo proceso de procesamiento de señales, y entorpece asimismo la tarea de detectar una resistencia del sistema en el sentido de que su comportamiento no se vea afectado por variaciones en el valor de algunos de los nodos de la red.

2.3.3. *Primera aproximación a un modelo dinámico*

En [17] se propone un sistema de ODEs no lineales, las cuales caracterizan el cambio en el tiempo de las concentraciones de las especies moleculares (valores de los nodos) presentes en la red, ya sean proteínas o ARNs. En base a esto, se realizan simulaciones temporales para medir el comportamiento del sistema. Un concepto interesante de este modelo es que los valores de los parámetros no son estimados a priori, sino que mediante distintos métodos (búsqueda al azar, por ejemplo) se generan conjuntos de parámetros que luego son probados verificando que el modelo reproduzca ciertos patrones macroscópicamente observables del modelo biológico estudiado.

En este enfoque, se proponen tres medidas de robustez:

- Con respecto a la solución obtenida, es decir con respecto a las características funcionales del sistema. Se propone que una cierta

arquitectura de red es robusta cuando, al generar conjuntos de parámetros aleatoriamente, presenta una proporción de soluciones válidas[2] que se desvía de lo esperable en función del azar.

- Con respecto a los valores de los parámetros, es decir respecto a los valores de los ejes. Se propone que una red es robusta si, una vez obtenida una solución válida, esta admite un amplio rango de variación para sus parámetros en forma individual manteniendo la (única) funcionalidad esperada. Se sugiere que esto podría simular la resistencia a pequeñas mutaciones en los genes.
- Con respecto a los valores de concentraciones iniciales, es decir los valores iniciales de los nodos. Se propone que una red es robusta si mantiene su patrón de actividad al variar las concentraciones iniciales de las especies, lo cual podría representar la resistencia de la red al "ruido" del entorno. No se presenta un desarrollo profundo de este concepto debido a las características puntuales del modelo en cuestión.

Si bien estas medidas de robustez son acertadas en el modelo puntual del artículo, resultan insuficientes al aplicarlas a un modelo de transducción de señales como EnvZ/OmpR, modelado en [2], o PmrA/PmrB - PhoP/PhoQ, estudiado en [14] y modelado en el presente trabajo. De acuerdo a la distinción de nodos de entrada y nodos de salida realizada anteriormente, estos tipos de sistemas presentan distintos patrones de valores en sus nodos de salida en función del estímulo presentado, es decir en función de los patrones de valores presentes en los nodos de entrada.

En un trabajo posterior [17], se plantea una medida para determinar la flexibilidad de una red genética. En el modelo estudiado, coincidentemente con la red PmrA/PmrB-PhoP/PhoQ, resulta de interés determinar el poder de las soluciones obtenidas para reproducir distintos patrones de salida dependiendo del patrón inicial de concentraciones presentado (es decir, dependiendo del patrón de valores presente en los nodos iniciales). Se sugiere que esta flexibilidad es alcanzable evolutivamente gracias a la robustez intrínseca de la red, es decir que ambas propiedades sistémicas se encuentran evolutivamente acopladas. Sin embargo, no se orienta la búsqueda de soluciones a obtener modelos flexibles, sino que esto se reporta más bien como una propiedad obtenida colateralmente. Asimismo, no se asegura que los distintos patrones a reproducir sean exhaustivos ni mutuamente excluyentes.

Otro aspecto importante a destacar es que conocer la arquitectura de una red no es suficiente, sino que, además, es necesario conocer las diferentes expresiones de los genes regulados por un mismo factor de trascripción, esto se debe al hecho que este último puede incidir de manera diferente en la expresión de distintos genes modelados en una misma red genética.

---

[2] A lo largo del texto denominaremos solución válida a un conjunto de parámetros que hace que la red pueda reproducir un cierto comportamiento deseado, como por ejemplo una serie de valores de concentraciones en los nodos de salida al inicializarse los nodos de entrada en un valor específico.

3. Algoritmos Genéticos

3.1. Introducción

Los algoritmos genéticos son una familia de modelos inspirados en la teoría de la evolución enunciada por Darwin (1859).

> *La Teoría de la Evolución explica el origen y la transformación de los seres vivos como el producto de la acción de dos principios fundamentales: la selección natural y el azar. La selección natural regula la variabilidad de la recombinación y mutación aleatorias de los genes: toda la variedad que observamos en la naturaleza se basa en la capacidad de los seres vivos de producir copias de sí mismos, en que el proceso de reproducción actualiza muchas variantes, y en que, en la interacción con el ambiente, algunas de ellas son seleccionadas para sobrevivir y producir las copias subsiguientes [4].*

Los algoritmos genéticos se utilizan para problemas de optimización y su aplicación hoy en día está bastante difundida a distintos problemas, como ser control adaptativo, juegos, problemas de trasportes, optimización en base de datos, aprendizaje automático, etc. Estos algoritmos fueron propuestos e investigados originalmente por John Holland y sus alumnos en Michigan, 1975 ([11] y [13]).

3.2. Descripción general del algoritmo genético

De manera general podemos decir, que los algoritmos genéticos trabajan con varias soluciones a la vez en las que se seleccionan las mejores y a través de recombinaciones y mutaciones se van generando nuevos conjuntos de soluciones.

Cada uno de estos conjuntos serán llamados *población* y cada una de las soluciones pertenecientes a éstos serán individuos, y estarán representadas por una codificación a la que denominaremos *cromosoma*. Esta codificación es dependiente del problema y es una de las primeras definiciones que se tienen que realizar para implementar un algoritmo genético. Por otro lado, cada cromosoma podrá estar formado por distintos componentes a la que llamaremos *genes*, de manera análoga a lo que sucede en los organismos vivos. Los valores que estarán dados por esta codificación lo llamaremos *fenotipo*, de la misma manera que sucede en la naturaleza, tenemos por un lado un *genotipo* que es la información contenida en los genes y por el otro un *fenotipo*, que es el resultado de la expresión de dicho genotipo.

Por otro lado, para poder determinar si un individuo es mejor que otro, necesitaremos definir una manera de calificar a cada solución, para ello se utiliza una *función de aptitud (fitness)*, la cual toma como entrada la información codificada en el cromosoma y devuelve un valor numérico. Esta función también es dependiente del problema.

Así como en la naturaleza los individuos se reproducen para formar nuevas generaciones, en los algoritmos genéticos llamaremos *generación* a cada iteración del algoritmo en la que individuos seleccionados para reproducirse generarán descendientes, los cuales a su vez también sufrirán el mismo proceso de selección y reproducción para producir nuevas generaciones. Otra de las definiciones que se deberá hacer cuando se diseña un algoritmo genético es el del tamaño de la población, el cual será constante de generación en generación.

Ahora, que hemos definido los principales conceptos que intervienen en los algoritmos genéticos, vamos a explicar, en las próximas secciones, cada uno de ellos de manera más detallada.

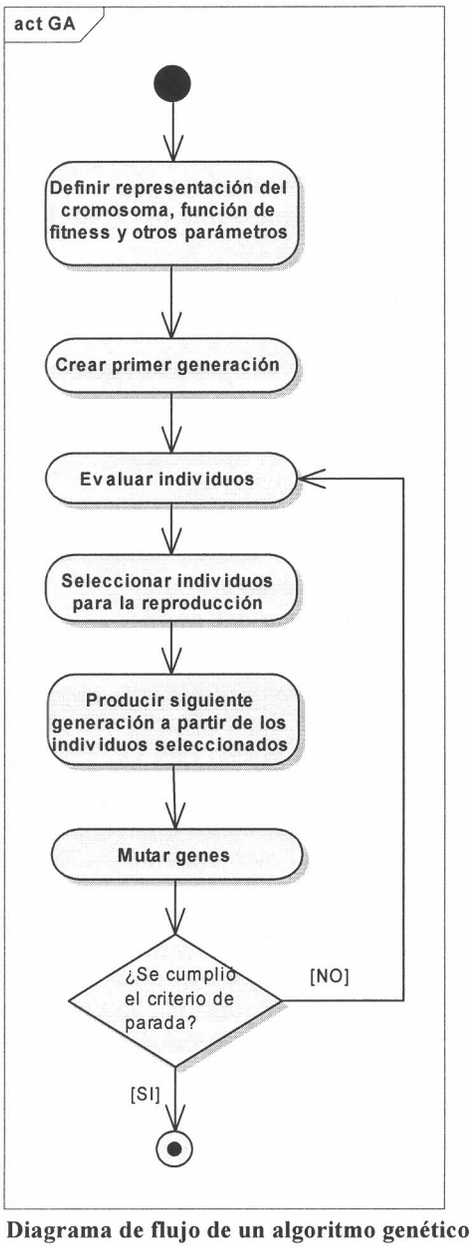


Figura 2.1. Diagrama de flujo de un algoritmo genético genérico

3.3. Codificación de las soluciones

Como se vio anteriormente, la codificación de las soluciones (cromosoma) es dependiente del problema a tratar e incide de manera directa sobre los resultados que se obtendrán en las ejecuciones del algoritmo genético. Existen muchos esquemas posibles de codificación, ya que los algoritmos genéticos pueden ser utilizados en distintos dominios. Dentro de estos podemos destacar los siguientes:

- **Codificación binaria**: Fue la primera en ser propuesta por Holland (1975) y está definida como una cadena de bits en la que cada gen puede estar formado por una subcadena de uno o más bits. ([11] y [13])
  En este tipo de representación necesitaremos una fórmula para determinar el cromosoma a partir del valor o los valores numéricos y otra para decodificarlo. Por ejemplo si queremos representar el valor 235, lo podríamos hacer con su codificación binaria, así tendríamos: 11101011.
  Para el caso de números reales deberíamos transformarlos en valores discretos, lo cual es un arte en sí mismo, ya que este proceso deberá tener en cuenta aspectos del problema que se intenta resolver.

- **Codificación real**: La codificación binaria presenta ciertas limitaciones cuando se trabaja con problemas que incluyen variables definidas sobre dominios continuos, como ser excesiva longitud de los cromosomas, falta de precisión, etc. Una posible manera de evitar estos inconvenientes es considerar un esquema de representación real. Aquí, cada variable del problema se asocia a un único gen que toma un valor real dentro del intervalo del problema, por lo que no existen diferencias entre el genotipo y el fenotipo. Es decir, no es necesario codificar ni decodificar los genes.
  Por ejemplo, si quisiéramos optimizar la función:

  $$f(x, y) = x.sin(4x) + 1,1y.sin(2y), \quad 0 \leq x \leq 10 \; y \; 0 \leq y \leq 10$$

  Podríamos representar al cromosoma como [x, y], en donde x e y son valores reales

En el método que se mostrará en la presente tesis se utilizará para codificar las soluciones este último esquema.

3.4. Primera generación

Para establecer la primera generación en la implementación de un algoritmo genético es muy común seleccionarlos al azar, es decir generando un conjunto de N cromosomas, siendo N el tamaño seleccionado de población para cada generación, eligiendo los valores de sus genes de manera aleatoria dentro del rango en que éstos son válidos.

3.5. Mecanismos de selección

El principal objetivo del operador de selección es producir una población intermedia de padres que se cruzarán para formar la nueva generación. Esta población intermedia se formará a partir de copias de los mejores individuos, eliminando de la población original a los peores, asegurándose de mantener constante el tamaño de la siguiente generación.

Se han propuesto diversas maneras de realizar estas acciones, las cuales se explican a continuación:

- **Selección por Torneo**: En esta estrategia de selección se toman N individuos y se los hace competir entre sí, de tal manera que el que tenga mejor valor de aptitud será el ganador. Esta acción se repite tantas veces hasta que el tamaño de esta población intermedia sea igual a la de la original. Este método ha sido muy utilizado debido a su simplicidad de implementación.

  Para el caso del *Torneo binario*, cada competencia se hace entre 2 individuos, lo cual requiere que cada uno compita dos veces. Esta estrategia asegura que el mejor individuo de la población ganará las dos competencias que dispute, con lo que aparecerá dos veces en la población intermedia y el peor ninguna, ya que perderá en sus dos contiendas.

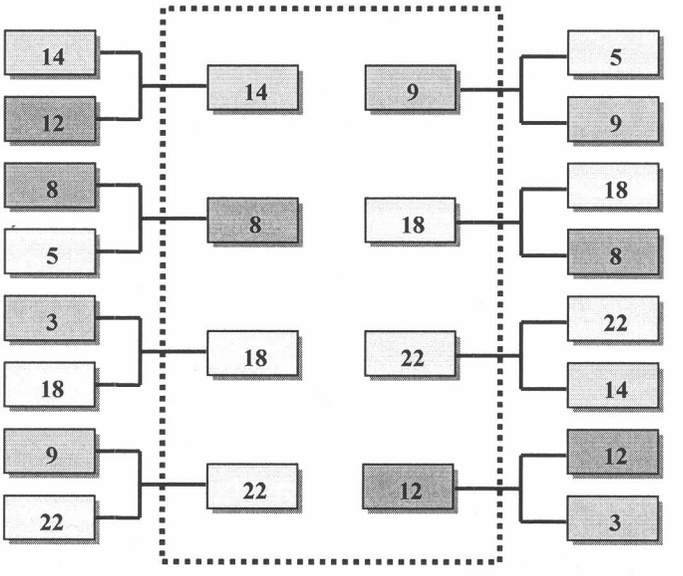


Figura 2.2. Ejemplo de Selección por Torneo Binario

- **Selección por Ruleta**: En este caso a cada individuo se le asigna una proporción o probabilidad de ser seleccionado en función a su valor de aptitud. El nombre de este método se debe a que esta acción puede ser vista como la formación de una ruleta que se divide en N porciones (siendo N el tamaño de la población) y el tamaño de cada una dependerá de la proporción de cada individuo

en función al valor de aptitud. Luego la ruleta se gira N veces y se toma el individuo que salga.
Debido a que los mejores individuos tienen una mayor proporción de la rueda, se espera que estos sean elegidos más veces que los peores en este método.

Individuo	Aptitud	Proporción
1	8	0,76
2	14	1,33
3	3	0,29
4	12	1,14
5	18	1,71
6	9	0,86
7	5	0,48
8	22	2,10
9	10	0,95
10	4	0,38

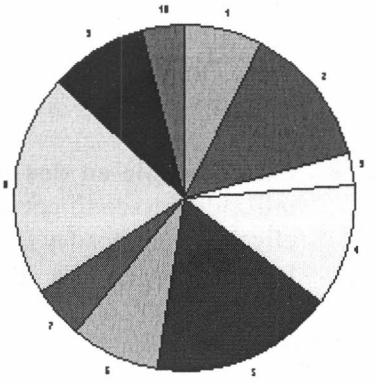


Figura 2.3. Ejemplo de selección por Ruleta

3.6. Operadores genéticos

Si solo se utilizara el operador de selección para producir las nuevas generaciones no sería posible construir nuevas soluciones, con lo que son necesarios otros operadores para realizar esto. Ellos son los operadores de cruce y de mutación.

El operador de cruce (*Cross Over*) es el mecanismo por el cual se comparte la información entre cromosomas, ya que combina la información de dos padres para producir dos hijos. Este proceso aplicado aleatoriamente en la población seleccionada para reproducirse permite obtener una nueva generación con la misma cantidad de individuos que la generación que le dio origen.

Al igual de lo que sucede con los operadores de selección, existen varios métodos de realizar este cruce. Además, debido a las diferentes estrategias de codificación de cromosomas, también es lógico que existan diferencias. Así tendremos:

- **Cruce simple en un punto**: Este operador es utilizado en codificaciones binarias o reales del algoritmo y consiste en tomar aleatoriamente un número i (punto de cruce), $1 \leq i \leq n$, siendo n la cantidad de bits que contiene el cromosoma e intercambiar los genes $i+1,\ldots,n$ genes siguiente entre los padres, como se muestra en el ejemplo siguiente:

```
Padre 1:   1 0 0 1 1 1 0 1 0 0 1
Padre 2:   1 1 0 1 0 1 1 0 0 1 0
```

Sea i = 5, entonces se generarían los siguientes hijos:

```
Hijo 1:   1 0 0 1 1 1 1 0 0 1 0
Hijo 2:   1 1 0 1 0 1 0 1 0 0 1
```

- **Cruce simple en dos puntos**: Este operador también puede ser utilizado en codificaciones binarias o reales. En este caso se eligen 2 números i y j, $1 \leq i < j \leq n$, intercambiando los genes de los padres en las posiciones $i+1...j$, como se muestra a continuación:

```
Padre 1:   1 0 0 1 1 1 0 1 0 0 1
Padre 2:   1 1 0 1 0 1 1 0 0 1 0
```

Sea $i = 4$ y $j = 8$, entonces se generarían los siguientes hijos:

```
Hijo 1:   1 0 0 1 0 1 1 0 0 0 1
Hijo 2:   1 1 0 1 1 1 0 1 0 1 0
```

Para los casos de codificaciones reales, se podría tomar cualquier cantidad de puntos de cruce, llegando incluso a tomarse tantos como genes tenga el cromosoma y de esa manera, seleccionar aleatoriamente cual de los padres contribuirá con un gen en cada posición del cromosoma.

```
Padre 1:   [P1, P2, P3, P4, ..., Pn]
Padre 2:   [M1, M2, M3, M4, ..., Mn]

Hijo 1:    [M1, P2, M3, M4, ..., Pn]
Hijo 2:    [P1, M2, P3, P4, ..., Mn]
```

- **Blend Cross Over**: El método anterior tiene como desventaja que no se agrega información nueva para cada valor de cada gen, ya que cada gen que fue generado aleatoriamente en la generación inicial se propaga en las sucesivas generaciones. Para solucionar este problema se propuso el método de *blending* el cual introduce nuevos valores para cada gen h_i de uno de los hijos a partir de la siguiente fórmula:

$$h_i = \beta p_i + (1 - \beta) m_i \qquad (2.1)$$

Siendo:

β: un número aleatorio entre 0 y 1.

p_i: el i-ésimo parámetro de cromosoma de un padre.

m_i: el i-ésimo parámetro de cromosoma del otro padre.

El mismo parámetro para el otro hijo se calcula con el complemento del primero (reemplazando β por $1 - \beta$). Si $\beta = 1$ entonces uno de los hijos tendrá un gen de unos de los padres, mientras que el otro el del otro padre. Lo mismo sucede si $\beta = 0$.

- **Linear Cross Over**: Si bien con el método anterior se incorporan nuevos valores para los genes, solo se agregaran valores acotados por los valores de los padres. El método lineal permite agregar valores que sean menores o mayores a los valores de los padres, con la diferencia adicional, que para lograr esto se generan tres hijos a partir de dos progenitores, en la que para cada gen *h*, los hijos estará formados por:

$$\begin{aligned} h_1 &= 0,5p + 0,5m \\ h_2 &= 1,5p - 0,5m \\ h_3 &= -0,5p + 1,5m \end{aligned} \qquad (2.2)$$

  Siendo p y m los genes de los correspondientes padres. Luego se seleccionan los dos mejores hijos para pasar a la siguiente generación

  Existen muchos otros métodos de cruce para valores reales, los cuales pueden verse en la bibliografía([11], [13]).

Otra manera de incorporar variaciones a los genes que existen en la población es a través de la mutación, esto se logra seleccionando un coeficiente de mutación, el cual es la probabilidad de que un gen mute dentro de la población. Habitualmente se utilizan coeficientes del orden del 1% al 15%. Una vez que un gen se ha seleccionado para ser mutado, se le modifica su valor dentro del rango en el que dicho gen es válido. Por ejemplo, si un gen binario tiene un valor igual a 1, se le reemplaza por un 0.

3.7. Elitismo

Puede suceder que al realizar el cruce la descendencia nunca supere a alguno de los padres, es por ello que se propuso una política más que es el elitismo. Este, lo que propone, es tener en cuenta la vida que tiene un individuo y de esa manera mantenerlo por más de una generación. Una forma simple de incorporar elitismo en la ejecución del algoritmo es la de mantener a los n mejores individuos (siendo n un porcentaje de la población) en la siguiente generación sin alterarlos de ninguna manera. Esto aumenta la convergencia del algoritmo, ya que estos mejores

individuos no se pierden, salvo que en alguna generación haya mejores que lo suplanten en las siguientes.

4. Problemas Multiobjetivo

Los algoritmos genéticos han demostrado una muy buena respuesta frente a problemas que requieren optimizar varios objetivos a la vez. Este tipo de problema se define de la siguiente manera:

$$
\begin{array}{lll}
\textit{maximizar/minimizar} & f_m(x), & m = 1, 2, \ldots, M; \\
\textit{sujeto a} & g_j(x) \geq 0, & j = 1, 2, \ldots, J; \\
 & h_k(x) = 0, & k = 1, 2, \ldots, K; \\
 & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \ldots, n.
\end{array}
$$

Siendo M el número de funciones objetivo que tiene el problema, J la cantidad de restricciones de desigualdad y K el número de restricciones de igualdad.

Una solución x es un vector de n variables de decisión: $x = (x_1, x_2, \ldots, x_n)^T$, donde cada variable x_i sólo puede tomar un valor dentro del intervalo $[x_i^{(L)}, x_i^{(U)}]$.

Se dice que una solución x domina a otra solución y cuando se cumplen las siguientes condiciones:

- La solución x no es peor que y para ningún objetivo.
- La solución x es estrictamente mejor que y en, al menos, un objetivo.

Si alguna de las condiciones anteriores es violada, la solución x no domina a la solución y. Si x domina a la solución y también es común escribir que x es no dominada por y.

Este concepto de dominancia es muy utilizado en diversos métodos de optimización multiobjetivo. Tomemos como ejemplo que tenemos dos funciones a optimizar, f a minimizar y g a maximizar y tomemos por ejemplo la siguiente tabla de posibles soluciones:

Sol.	f(x)	g(x)
1	12	5
2	18	2
3	7	4
4	8	3
5	16	14

Aquí podemos ver que la solución 1 es mejor que la solución 2 para ambos objetivos, ya que $f(x_1) < f(x_2)$ y $g(x_1) > g(x_2)$, con lo que podemos decir que la solución 1 domina a la 2.

Tomemos ahora la solución 3, vemos que la función f de la solución 3 es mejor que la de la solución 1, pero no sucede lo mismo con la función g, en la que la solución 1 es mejor. En este caso no podemos decir que una

solución es mejor que la otra, por lo que se suele decir que la solución 1 y la 3 son no dominadas una respecto a la otra.

Siguiendo con este método podríamos comparar todas las soluciones entre si y determinar cuales soluciones dominan a otras y cuales no son dominadas, de esa manera llegaremos a obtener un conjunto de soluciones que no son dominadas entre si, el cual tiene la propiedad que cualquier solución que no pertenezca a él es dominada por al menos un elemento de este conjunto. Esto equivale a decir que cualquier otra solución que no pertenezca a este conjunto es peor que las que pertenecen a este conjunto.

No es extraño que en este tipo de problemas no aparezcan soluciones factibles que sean óptimas simultáneamente para todos los objetivos. En este caso, la solución matemática más adecuada es quedarse con aquellas soluciones que ofrezcan el menor conflicto posible entre objetivos.

Para lograr esto, todas las técnicas clásicas reducen el vector objetivo a un escalar, es decir, a un único objetivo. En estos casos, en realidad, se trabaja con un problema sustituto buscando una solución sujeta a las restricciones especificadas. Entre estas técnicas vamos a destacar la Optimización Mediante Ponderación de los Objetivos o sumas pesadas, que es, probablemente, la técnica más simple. En este caso, las funciones objetivo se combinan en una función objetivo global, F, de la siguiente manera [13]:

$$F(x) = \sum_{i=1}^{M} w_i f_i(x) \qquad (2.3)$$

donde w_i se define como:

$$F(x) = \sum_{i=1}^{M} w_i = 1 \quad ; \quad \forall w_i \in \mathbb{R},\ 0 \leq w_i \leq 1 \qquad (2.4)$$

En este método se utiliza un vector de pesos, los cuales controlan la relevancia de cada objetivo, con lo que modificando estos valores podemos modificar dicha relevancia.

5. Comentarios Finales

En este capítulo hemos estudiado los principales conceptos computacionales relacionados a las redes genéticas y a los modelos de optimización que utilizaremos para desarrollar el método propuesto en este trabajo (Algoritmos Genéticos).

Con respecto al primero hemos visto que existen numerosas propuestas para modelar las redes genéticas y ellas están basadas principalmente en lo que se intenta estudiar de estas redes, así tenemos modelos estáticos o dinámicos, determinísticos o probabilísticos, con valores booleanos, discretos o continuos, y según sea la aproximación serán modelos más o menos complejos. La decisión de cual modelo a usar estará dada por distintos aspectos a tener en cuenta, como ser:

- ¿De cuánta información disponemos?

- ¿Cuántas especies están presentes en la red?
- ¿Qué esperamos predecir con el modelo elegido?
- ¿Cuán complejo, computacionalmente, es el modelo elegido?

Por otro lado, ninguno de los modelos analizados propone un método que permita la formalización automática de un problema biológico mediante la generación de arquitecturas de red que satisfagan las reglas biológicas que lo definen. Tampoco se propone un mecanismo que permita reformular estas reglas, es decir generar nuevas hipótesis sobre el sistema, en función de los resultados obtenidos. Veremos en los próximos capítulos una propuesta para lograr estos objetivos.

Para este trabajo, hemos elegido la opción de modelar a los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ de Salmonella Enterica, mediante un sistema dinámico de ecuaciones diferencias ordinarias (ODEs), el cual se describirá en mayor detalle en los siguientes capítulos.

En cuanto a los algoritmos genéticos, hemos visto que constituyen una aproximación a la optimización de diversos problemas, ya que existen diversas opciones de implementación de sus operadores, lo cual lo hace muy flexible para resolver, incluso, problemas multiobjetivos. En el capítulo 4 explicaremos en detalle la implementación de un algoritmo genético que se utilizó para la determinación del conjunto de parámetros iniciales para la ejecución de la red genética que describe la dinámica de los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ.

Capítulo 3

Aproximación por búsqueda aleatoria al problema del aprendizaje de redes regulatorias genéticas

1. Introducción

Como se explicó en el capítulo anterior, el objetivo de la definición de redes genéticas es la comprensión de la dinámica entre los genes y sus productos para permitir la expresión o inhibición de otros genes y de esa manera determinar características que permitan, por ejemplo, desarrollar nuevas drogas.

En este capítulo mostraremos un modelo existente, basado en búsquedas aleatorias, para la inferencia de redes genéticas basado en un análisis secuencial e iterativo de ciertas propiedades que deberían poseer las arquitecturas para determinar su calidad, que realiza búsquedas por un mecanismo aleatorio [22]. En cada una de las secciones se analizará cada uno de los elementos que constituyen la metodología de trabajo, siendo estos:

- La identificación de reglas en base a sitios de vinculación y la bibliografía disponible.
- La identificación de arquitecturas consistentes con estas reglas a partir de la generación de conjuntos de parámetros mediante un método de búsqueda aleatoria.
- La evaluación de realismo, flexibilidad y robustez de las arquitecturas obtenidas.
- La reformulación de las mismas, generando hipótesis de una manera adaptativa.
- La realización de predicciones .
- La contrastación experimental biológica.

2. El método

2.1. Introducción

En los enfoques expuestos en el capítulo anterior se presentan distintas medidas para determinar la robustez y flexibilidad de una red genética en base a modelos computacionales. Sin embargo, estos enfoques son

parciales y no presentan una medida integral acerca de la "calidad" de la red propuesta para representar un modelo biológico. Por tal motivo, este método propone una medida de calidad, basada en los criterios que se siguen a continuación:

- **Realismo**: la red debe ser capaz de reproducir el comportamiento observable experimentalmente, es decir el funcionamiento biológico de los sistemas en estudio. Una red realista en este sentido debe ser capaz de reproducir este funcionamiento en forma relativamente independiente de los parámetros escogidos (por ejemplo, al generar aleatoriamente conjuntos de parámetros, la red debe reproducir el comportamiento biológico en un proporción que se desvíe de lo esperable en función del azar).
- **Robustez** respecto de
  - **parámetros (ejes)**: la red debe soportar variaciones de distinto rango en parámetros particulares, sin que se vea afectado su realismo. Esto representa la propiedad biológica de resistencia a pequeñas mutaciones en los genes de la red.
  - **concentraciones (nodos)**: la red debe soportar variaciones en los valores de los nodos de entrada, sin que se vea afectado su realismo. Esto representa la resistencia de la red al ruido presente en los sistemas moleculares.
- **Flexibilidad**: la red debe ser capaz de reproducir simultáneamente los distintos patrones de comportamiento observados (o propuestos) en los sistemas biológicos en estudio. Este criterio se encuentra íntimamente relacionado con la robustez, en el sentido de que es plausible proponer que la flexibilidad comúnmente observada en las redes genéticas es alcanzable en sistemas robustos que soporten la acumulación de mutaciones que producirá, a lo largo de la evolución, el surgimiento de nuevos patrones funcionales.

2.2. Flujo de ejecución del método

El flujo de la ejecución del algoritmo basado en búsquedas aleatorias puede resumirse mediante el diagrama de flujo que se presenta en figura 3.1, en la cual se muestran los principales procesos involucrados en la generación, evaluación y regeneración de las arquitecturas propuestas para modelar una red genética determinada. En los distintos puntos de control del diagrama se muestran las etapas del algoritmo en las que se realizan pruebas de ciertas propiedades requeridas por la red genética, las cuales son realismo, flexibilidad y robustez, según se vio en la sección anterior.

En el presente trabajo solamente nos concentraremos en los pasos 1 a 5 del flujo de la figura 3.1, ya que es en ese aspecto que intentaremos optimizar las soluciones obtenidas con el método que presentaremos en el próximo capítulo.

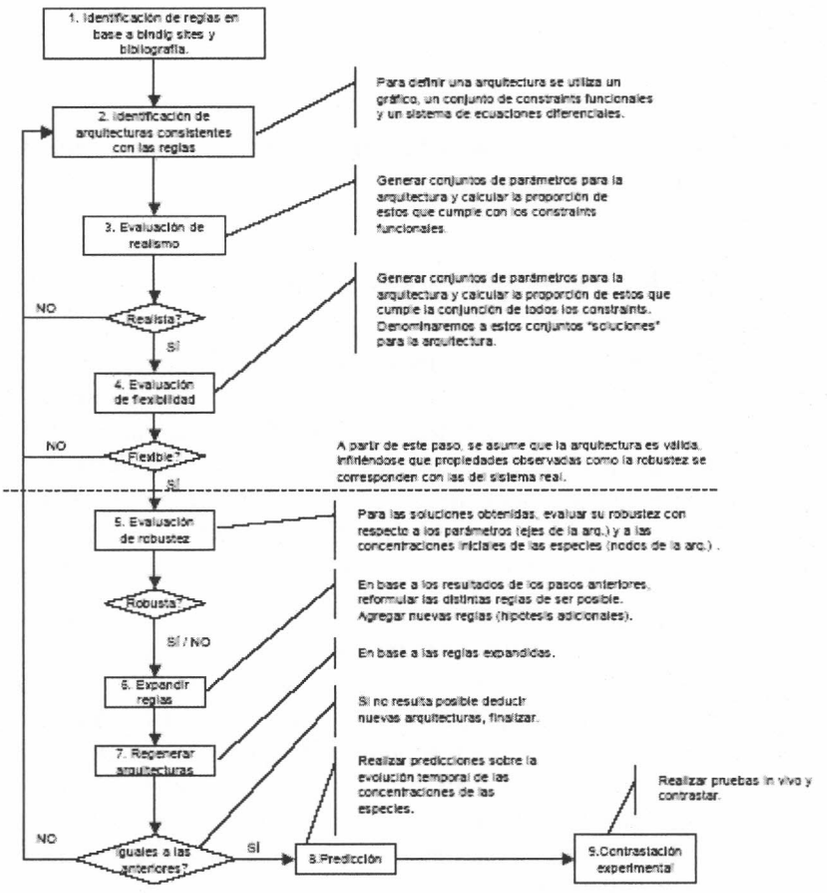


Figura 3.1: Diagrama de flujo para el algoritmo basado en búsquedas aleatorias. La numeración de los distintos procesos del diagrama se corresponde con la utilizada en los pseudoalgoritmos para identificar a las distintas funciones.

2.3. Identificación de reglas y de arquitecturas consistentes con ellas

En base a las reglas para el funcionamiento de una red genética recogidas de la literatura y la evidencia de los sitios de vinculación de los genes involucrados en dicha red, es posible proponer una o varias arquitecturas para los sistemas que se intentan estudiar. Estas arquitecturas modelan las interacciones entre las especies, y pueden representarse como grafos en los cuales las especies involucradas (proteínas y ARN) se visualizan como nodos, y los ejes representan las interacciones entre las ellas.

Estas arquitecturas pueden obtenerse a partir de librerías de módulos funcionales como los descriptos en [15], [18] y [12], definiéndose a la red como una agregación de submódulos, que representan distintas interacciones entre los componentes de la red, entre ellas se pueden destacar:

- Mecanismos de autorregulación de un gen (positiva o negativa)
- Regulación de un gen en base a los productos de un grupo de genes

- Fosforilación y defosforilación de una proteína mediada por actividad enzimática

Para el caso del método que se está presentando en este trabajo, las propiedades que lo caracterizan son:

- A partir de los distintos submódulos de la red genética, se utilizan librerías de ecuaciones diferenciales, como las enumeradas en [16], [23] y [12], para la transformación en sistemas de ODEs. De esta manera, el valor de los ejes de la arquitectura derivada queda determinado por las ecuaciones propuestas como una función de ciertos parámetros bioquímicos. Estos parámetros son en general desconocidos, deben ser estimados de manera computacional.

  También existen enfoques para la inferencia de arquitecturas que utilizan evidencia experimental para fijar los parámetros [2], pero por la naturaleza de estas redes, en la que la cantidad de parámetros es muy grande, se hace imposible determinarlas experimentalmente, incluso aunque la red no tenga un tamaño tan importante. Por otro lado, esta determinación experimental puede resultar perjudicial dado que impide extraer información acerca del papel de las variaciones que se pueden presentar dentro del funcionamiento de la red.

- Las ecuaciones obtenidas para los diferenciales en función del tiempo de las especies permiten calcular, en un tiempo dado, la concentración de una especie (valor de un nodo) como una función de los valores de las concentraciones de las especies (nodos) conectadas con ella. Esto permite la realización de simulaciones temporales sobre el sistema de ODEs obtenido, posibilitando capturar el comportamiento dinámico del sistema.

- Tanto los parámetros utilizados como las concentraciones son adimensionales, por lo que no resulta necesario realizar normalizaciones de unidades en las ecuaciones. Las concentraciones de las especies toman valores reales entre 0 (nulo) y 1 (máximo). Las conversiones algebraicas utilizadas para lograr estos valores adimensionales se describen en [23].

2.4. Evaluación de realismo

Con el fin de evaluar la capacidad de una arquitectura en estudio para reproducir el comportamiento real de la red genética en organismos vivos se realizan sucesivas pruebas. El modelo general para las mismas consiste primeramente en, dada una cierta arquitectura, proponer concentraciones iniciales para las especies representadas en el modelo según las determinadas en experimentos biológicos. Luego se realizan iterativamente asignaciones aleatorias de valores a los parámetros (*búsqueda aleatoria*) dentro de los rangos de valores biológicamente

válidos (en base a lo propuesto en [23]), seleccionando aquellos conjuntos que al simular el comportamiento de la red reproducen las funcionalidades pedidas, conjunto al que llamaremos *solución de la red* o simplemente *solución*. En cada una de estas simulaciones la red obtiene un puntaje (*score*) el cual debe ser menor a un valor umbral para decir que dicha arquitectura cumple con las condiciones dadas.

Una vez que se ha realizado una cantidad de simulaciones, se determina la proporción de soluciones encontradas en relación a la cantidad de iteraciones que se ejecutaron. Diremos que la arquitectura es realista si esta proporción supera a una proporción esperada, la cual está determinada previamente. En la siguiente figura mostramos el pseudoalgoritmo de este proceso.

```
CONSTANTES:
   cantItentos: Cantidad de iteraciones que tendrá el
   algoritmo
   propAceptacion: Proporción mínima para considerar que la
   arquitectura es realista

FUNCIONES AUXILIARES:
   correrSimulación(arq, p, f): Función que devuelve
   verdadero si el conjunto de parámetros p cumplen con la
   funcionalidad f para la arquitectura arq. Falso para el
   caso contrario
   generarParametros(arq): Genera aleatoriamente el
   conjunto inicial de parámetros para la arquitectura arq

INPUT:
   arq: Arquitectura a evaluar.
   funcs: Conjunto de funcionalidades a evaluar para la
   arquitectura arq.

RESULTADO:
   ret: Verdadero si la arquitectura cumple con las
   funcionalidades funcs. Falso en caso contrario

verificarRealismo(arq, funcs)
   ret = True
   for f in funcs do:
      cant = 0
      for 1 to cantIntentos do:
         p <- generarParametros(arq)
         cumple <- correrSimulacion(arq, p, f)

         if (cumple) then:
            cant++
         else:
            skip

      ret <- ret & (cant / cantIntentos ≥ propAceptacion)

   return ret
```

Figura 3.2. Pseudoalgoritmo del proceso de determinación del realismo de una arquitectura.

Este esquema de búsqueda permite calcular la frecuencia de soluciones válidas obtenidas, y en base a ésta calcular la probabilidad de encontrar por azar un conjunto que reproduzca el patrón pedido. Si bien en el algoritmo sólo se utiliza la frecuencia, una alta probabilidad de encontrar aleatoriamente conjuntos de parámetros que hagan que la arquitectura cumpla con las funcionalidades pedidas puede indicar que esta funcionalidad es más dependiente de la arquitectura en si misma que de los parámetros escogidos. Si consideramos que p es probabilidad de escoger aleatoriamente un valor para un parámetro que confiera a la arquitectura la habilidad de reproducir una funcionalidad, f la frecuencia de soluciones encontradas y n la cantidad de parámetros, se cumple que:

$$p^n = f \qquad (3.1)$$

de esta manera, la probabilidad de encontrar aleatoriamente un conjunto de parámetros que constituyan una solución válida[3] para la arquitectura puede calcularse como:

$$p = 10^{\frac{\log f}{n}} \qquad (3.2)$$

Al realizar las pruebas de realismo sobre una arquitectura, se considera a la red genética que está modelada como una "caja negra" a la cual se le pide que reproduzca una cierta funcionalidad[4] en una cierta proporción de los conjuntos de parámetros generados aleatoriamente sin plantearse requerimientos adicionales acerca de otras funcionalidades que la arquitectura podría reproducir o no. El objetivo de las mismas es solamente determinar si la arquitectura es capaz de reproducir una cierta funcionalidad. Como puede verse en el código de la función utilizada, los conjuntos de parámetros utilizados para cada simulación son distintos, por lo que en este punto no puede determinarse si la arquitectura puede cumplir todas las funcionalidades planteadas con *un mismo conjunto de parámetros.*

Si bien este esquema podría parecer insuficiente para un problema genérico, los resultados obtenidos en [23] y [17] sugieren que para un problema como el estudiado la noción de soluciones globalmente óptimas carece de sentido biológico, y muestran una distribución de las soluciones en el espacio de búsqueda que disminuye significativamente la utilidad de algoritmos de búsqueda más refinados. La posibilidad de determinar el realismo de una arquitectura propuesta independientemente

---

[3] Un conjunto de parámetros es una solución válida para la arquitectura se al realizar una simulación sobre la misma con esos parámetros esta produce los valores requeridos por una cierta funcionalidad, o limitación.

[4] Se requiere que la arquitectura, al asignar ciertos valores de concentración a las especies distinguidas como "de entrada" en el tiempo T_0, obtenga valores de concentración específicos en los nodos de "salida" en un cierto tiempo de la simulación T_n posterior.

de los parámetros escogidos constituye una gran ventaja sobre los enfoques de modelado basados en ecuaciones diferenciales como el propuesto por Batchelor y Goulian en [2] para el sistema de dos componentes EnvZ/OmpR.

2.5. Flexibilidad y Completitud Funcional

Para evaluar la flexibilidad de las arquitecturas, se realizan nuevas búsquedas utilizando como criterio para aceptar una solución la conjunción de todas las condiciones propuestas. Esto es posible realizando simulaciones independientes para cada conjunto de parámetros generado, cada una conteniendo distintos valores iniciales de concentraciones de las especies. Se simulan así las distintas combinaciones de señales a las que se ve expuesta la red real. El cálculo de si la arquitectura es flexible se realiza mediante la siguiente función:

```
CONSTANTES:
   cantIntentos: Cantidad de iteraciones que tendrá el
   algoritmo
   propAceptacion: Proporción mínima para considerar que la
   arquitectura es realista

FUNCIONES AUXILIARES:
   correrSimulación(arq, p, f): Función que devuelve
   verdadero si el conjunto de parámetros p cumplen con la
   funcionalidad f para la arquitectura arq. Falso para el
   caso contrario
   generarParametros(arq): Genera aleatoriamente el
   conjunto inicial de parámetros para la arquitectura arq

INPUT:
   arq: Arquitectura a testear.
   funcs: Conjunto de funcionalidades a testear para la
   arquitectura arq.

RESULTADO:
   ret: Valor booleano que determina si la arquitectura
   cumple con la conjunción de las funcionalidades funcs.

verificarFlexibilidad(arq, funcs)
   cant = 0
   for 1 to cantIntentos do:
      p <- generarParametros(arq)
      cumple = True
      for f in funcs do:
         cumple <- cumple & correrSimulacion(arq, p, f)

      if (cumple) then:
         cant++
      else:
         skip

   ret <- cant / cantIntentos ≥ propAceptacion

   return ret
```

Figura 3.3. Pseudoalgoritmo del proceso de determinación de la flexibilidad de una arquitectura.

Un conjunto de parámetros es considerado una solución válida cuando los resultados de las simulaciones reproducen todas las funcionalidades requeridas. Se considera entonces que una arquitectura es flexible en su funcionalidad en base a la proporción de conjuntos valores de los parámetros generados aleatoriamente que le permiten reproducir todas las funcionalidades propuestas.

La determinación de las funcionalidades a cumplir no es en absoluto trivial, dado que estas deben cubrir todas las combinaciones de patrones de entrada - salida a las cuales la red genética modelada debe responder en el organismo real. Por ejemplo, debe asegurarse que las funcionalidades modelen el hecho de que si no se somete a la red a ningún estímulo, esta no debería reproducir ningún comportamiento específico, salvo que este sea el caso en la red genética estudiada. De esta manera, puede proponerse que una red que cumple con la propuesta flexibilidad funcional es necesariamente completa en su funcionalidad. Esta definición de flexibilidad funcional de una arquitectura, asociada al concepto de completitud funcional, constituye una gran ventaja sobre el enfoque propuesto para el modelado basado en ecuaciones diferenciales propuesto por Von Dassow et Al. en [23], en el cual la flexibilidad de las arquitecturas obtenidas no se analiza en forma exhaustiva.

En base a este procedimiento, es posible distinguir el subconjunto de arquitecturas "flexibles funcionalmente" dentro del conjunto de arquitecturas "realistas".

2.6. Robustez de los Parámetros

Una vez definida una medida para el realismo y la flexibilidad de las redes propuestas, surge el interrogante de si estas propiedades son dependientes de los parámetros específicos escogidos, o si son realmente intrínsecas a la topología, como se sugiere en secciones anteriores. Cabe preguntar entonces si, dada una solución válida para el sistema, esta deja de cumplir con el comportamiento pedido al modificar alguno de sus parámetros. Y en el caso de que una perturbación modifique el comportamiento, puede preguntarse cuánto es necesario perturbar este parámetro.

Para obtener una medida de la robustez, se toma una solución escogida aleatoriamente entre las que reproducen la conjunción de todos las limitaciones, y se calcula la puntuación obtenida por la red al modificar un parámetro particular de una solución (y dejando los demás fijos), haciendo que este tome valores a lo largo de todo el intervalo definido como biológicamente significativo. Este proceso se repite para todos los parámetros. A continuación se muestra un pseudoalgoritmo para dicho proceso.

```
CONSTANTES:
   cantPuntos: Cantidad de puntos a tomar dentro del rango
   de valores para un parámetro de la red
```

```
   propRobustezParam: Al analizar el poder de la
   arquitectura para reproducir una funcionalidad dada, si
   esta se mantiene para esta proporción de los valores
   analizados, el parámetro se considera robusto.
   propParamsRobustos: La solución es considerada robusta
   con respecto a sus ejes si la proporción de parámetros
   robustos iguala o supera esta cantidad
   propSolucionesRob: La arquitectura es considerada
   robusta con respecto a sus ejes si la proporción de
   soluciones con parámetros robustos iguala o supera esta
   cantidad

FUNCIONES AUXILIARES:
   correrSimulación(arq, p, f): Función que devuelve
   verdadero si el conjunto de parámetros p cumplen con la
   funcionalidad f para la arquitectura arq. Falso para el
   caso contrario
   setParam(sol, p, punto): asigna el valor punto al
   parámetro p en el conjunto sol
   generarSeriePuntos(p, cantPuntos): Generar una serie de
   puntos equidistantes dentro del rango valido para el
   parámetro p

INPUT:
   arq: Arquitectura sobre la cual verificar robustez.
   funcs: Conjunto de funcionalidades a cumplir por la
   arquitectura arq.

RESULTADO:
   ret: Booleano que determina si la arquitectura es
   robusta con respecto a sus ejes.

robustezEjes(arq, funcs)
   soluciones <- soluciones(arq) //conjunto de conjuntos de
parámetros
                                 //que son soluciones para
la arq
                                 //calculado en un paso
anterior
   cantParams <- cantParams(arq) //cantidad de parámetros
en arq
   cantSolucionesRob <- 0
   ret <- True
   for sol in soluciones do:
      cantParamsRob <- 0
      for p in sol do:
         cantPuntRob <- 0

         serie <- generarSeriePuntos(p, cantPuntos)
         for punto in serie do:

            sol' <- setParam(sol, p, punto)
            esRob <- correrSimulacion(arq, sol', func)
            if esRob then: cantPuntRob++ else: skip

         if cantPuntRob / cantPuntos ≥ propRobustezParam
then:
            cantParamsRob++
         else:
            skip
```

```
      if (cantParamsRob / cantParams ≥ propParamsRobustos)
then:
         cantSolucionesRob++
      else:
         skip

   ret <- cantSolucionesRob / size(soluciones) ≥
propSolucionesRob

   return ret
```

Figura 3.4. Pseudoalgoritmo del proceso de determinación de la robustez de una solución.

2.7. Robustez de las Concentraciones Iniciales

El modelado efectuado sobre el problema biológico presenta limitaciones con respecto a los valores que se asignan a las especies del modelo, debido a que estos son fijados al comenzar la simulación mientras que en los sistemas biológicos una uniformidad tan estricta no parece razonable. En el caso de los valores de los parámetros de las ecuaciones diferenciales, esta problemática ya ha sido abordada. Sin embargo, resta analizar la robustez de las arquitecturas propuesta con respecto a los valores iniciales de las concentraciones de las especies.

Para medir el impacto de la variación de las concentraciones iniciales en la capacidad funcional de la red, se realizan simulaciones sobre soluciones obtenidas en los pasos anteriores, a las cuales se les han introducido cambios en los valores iniciales de las especies (nodos de la red). Luego se calcula la proporción de estas que conserva la funcionalidad deseada luego de las perturbaciones, considerándose que la arquitectura es robusta si esta proporción es mayor a un cierto umbral. El pseudoalgoritmo es el siguiente:

```
CONSTANTES:
   propSolsRobustasNodos: Proporción de soluciones robustas
   al variar los valores iniciales de los nodos a partir de
   la cual se considera robusta a la arquitectura

FUNCIONES AUXILIARES:
   correrSimulación(arq, p, f): Función que devuelve
   verdadero si el conjunto de parámetros p cumplen con la
   funcionalidad f para la arquitectura arq. Falso para el
   caso contrario
   setCondInic(arq, cond): asigna la condición inicial a la
   arquitectura

INPUT:
   arq: Arquitectura sobre la cual verificar robustez.
   funcs: Conjunto de funcionalidades a cumplir por la
   arquitectura arq.

RESULTADO:
   ret: Booleano que determina si la arquitectura es
   robusta con respecto a sus nodos.
```

```
robustezNodos(arq, funcs)
   soluciones <- soluciones(arq) //conjunto de conjuntos de
                                 //parámetros que son
solución para la
                                 //arquitectura calculado
en un paso
                                 //anterior

   //Generar un conjunto de conjuntos de valores iniciales
   //para los nodos.
   condIniciales <- generarCondInic(arq)
   cantSolsRob <- 0

   for sol in soluciones do:
      for cond in condIniciales do:
         arq' <- setCondInic(arq, cond)
         esRob <- correrSimulacion(arq', sol, func)
         if esRob then: cantSolsRob++ else: skip

   ret <- (cantSolsRob / (#soluciones * #condIniciales))
           ≥ propSolsRobustasNodos

   return ret
```

Figura 3.5. Pseudoalgoritmo del proceso de determinación de la robustez de las concentraciones iniciales para una solución.

Es importante destacar que en todos los casos se trata de modificaciones "suaves" sobre los sistemas. Efectivamente, el problema biológico estudiado corresponde a un sistema de transducción de señales, y por lo tanto no resultaría coherente esperar un comportamiento uniforme de la red al modificar excesivamente los valores que determinan la presencia o no de estas señales y de las respuestas del sistema.

3. Comentarios Finales

En este capítulo se enunciaron los elementos que permiten formular una metodología basada en búsqueda aleatoria para el análisis de redes genéticas. Esta permite transformar la evidencia de los sitios de vinculación (*binding sites*) a los genes de la red genética modelada y la información existente en la bibliografía en un conjunto de reglas que determinan el comportamiento de la red. Estas reglas son luego modeladas mediante módulos arquitectónicos existentes en una librería de módulos, combinándose en una arquitectura completa. Esta arquitectura es implementada realizando un mapeo entre los módulos y las ecuaciones diferenciales existentes en una librería de ecuaciones existente. El sistema de ecuaciones diferenciales obtenido permite realizar simulaciones sobre el sistema, testeando propiedades tales como su realismo, flexibilidad y robustez. La validación o no de estas propiedades aporta importante información para la inferencia de nuevas hipótesis sobre el comportamiento del sistema. Este esquema iterativo de formulación, prueba y reformulación puede repetirse hasta obtener una arquitectura válida para el sistema estudiado, con el fin de realizar predicciones sobre el mismo. Estas predicciones permitirán dirigir la

experimentación biológica, acelerando el proceso de exploración de las redes genéticas estudiadas.

Sin embargo, este método tiene como principal desventaja que utiliza una aproximación por búsqueda aleatoria en la que para cada iteración genera un nuevo conjunto de parámetros al azar para comenzar la simulación, esto hace que, debido a las características continuas de los parámetros y a la cantidad de parámetros que usualmente tienen las redes, el método sea muy ineficiente para encontrar soluciones y además que la probabilidad de encontrar las mejores sea muy baja. En el siguiente capítulo mostraremos como se mejora el método utilizando un algoritmo genético para la obtención de dichos parámetros.

Capítulo 4

Método Mejorado

1. Introducción

En el capítulo anterior vimos un método basado en búsqueda aleatoria para aprendizaje de redes genéticas, el cual se desarrolló en el contexto de una Tesis de Licenciatura del Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires[22]. También vimos que este método tenía la desventaja de obtener bajas proporciones de resultados, con un valor muy cercano al valor umbral (*threshold*). Recordemos que cuanto menor es el score obtenido en la simulación mejor es la solución. Para optimizar este método, son dos los problemas a ser considerados. El primero de ellos es el de obtener las condiciones iniciales para que la corrida de la simulación sea exitosa, es decir, que el conjunto de parámetros de la red posea valores que aseguren que el puntaje final de la corrida (*score)* sea menor que cierto valor umbral determinado (*threshold*). El otro de los problemas es el de obtener resultados que sean biológicamente significativos.

Nosotros proponemos un método que consiste, en lugar de generar una configuración de parámetros inicial al azar para cada simulación, como en el caso del método visto en el capítulo anterior, en optimizar la generación de soluciones a partir de un algoritmo genético, que asegurará la obtención de mayor proporción de soluciones en función a las simulaciones corridas, con un mejor *score* para las mejores de ellas, en relación a las que se obtienen con el método basado en búsqueda aleatoria.

En las secciones siguientes explicaremos la forma en la que enfrentamos el primer problema a partir de la implementación de un algoritmo genético, dejando el otro para el final del capítulo.

2. El método

El método desarrollado se puede ver esquemáticamente a partir del siguiente diagrama de actividades.

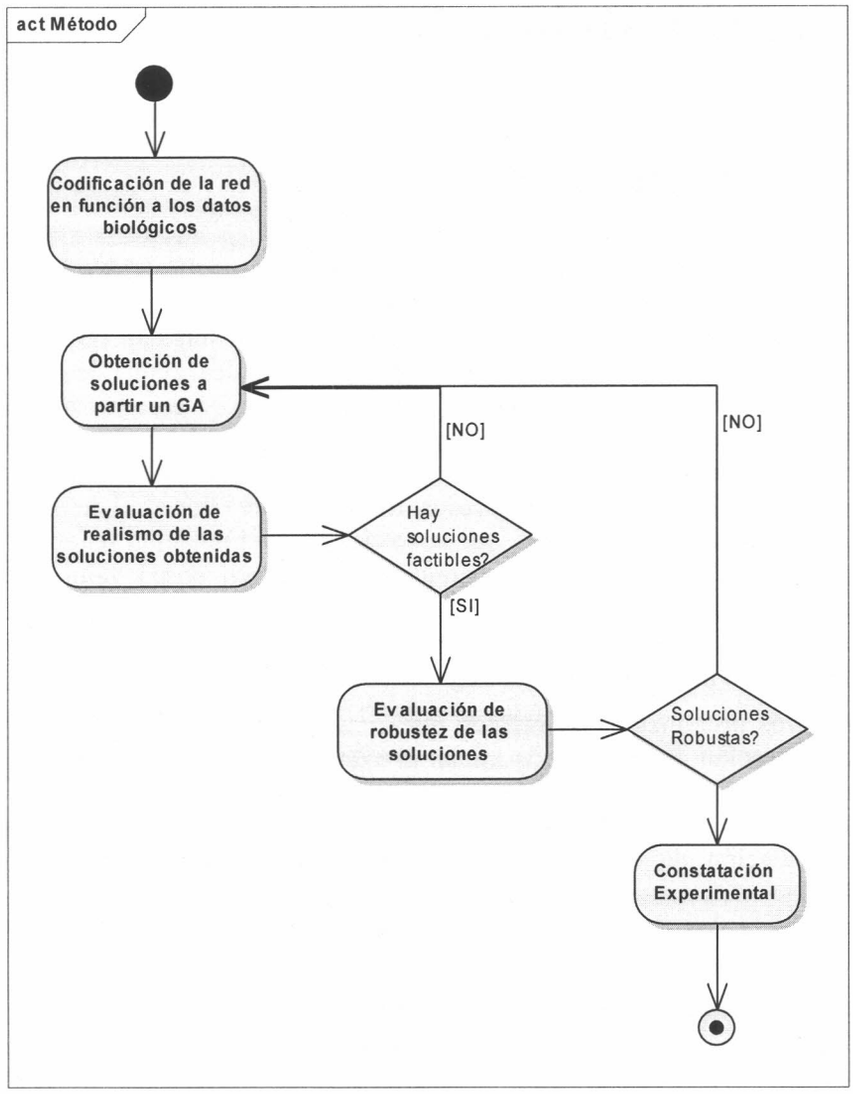


Figura 4.1. Diagrama de actividades que representa el método desarrollado.

En las siguientes subsecciones se presenta el detalle del método desarrollado.

2.1. Modelado de la red genética a partir de ODEs no lineales

Al igual que en el método explicado en el capítulo anterior, las arquitecturas de las redes genéticas se basan en las reglas recogidas de la literatura y en la evidencia de los sitios de vinculación de los genes involucrados en dicha red.

En las siguientes subsecciones se presentan dos ejemplos del mecanismo de construcción de una arquitectura a partir de interacciones entre los componentes de la red, siendo éstas *Retroalimentación (Feedback) Positiva de un Gen y Fosforilación-Defosforilación de una Proteína*, mediada por actividad enzimática. Antes de mostrar estos ejemplos, es importante aclarar que, si bien para una mejor comprensión del problema

se muestran las interacciones con especies del sistema PhoP/PhoQ, esta esquematización es válida en general.

Como convención, para denominar a las especies en los diagramas de la arquitectura, utilizaremos letras mayúsculas para los nombres de las proteínas y minúsculas para los ARNs. El ADN no se modela explícitamente, y por lo tanto la interacción de, por ejemplo, una proteína activadora sobre un cierto gen se muestra en la arquitectura mediante un eje entre la especie proteica y la especie ARN, simbolizando que la cantidad de ARN puede verse como una función de la cantidad de activador presente.

Las ecuaciones diferenciales utilizadas para modelar las reacciones contienen parámetros con distinta significación biológica. Estos parámetros serán el eje en nuestro método, ya que su valor definirá el éxito o fracaso de una corrida de la simulación de la red modelada por estas ecuaciones. Con el fin de aumentar la declaratividad de los parámetros, se utilizó la siguiente convención de nombres:

- Prefijo (ejemplo, H, nu): Identifica la función biológica del parámetro (vida media, coeficiente de cooperatividad de Hill, etc.)
- Nombre: nombre de las especies involucradas en la reacción.

Por ejemplo `nu_PHOQ_PHOP-P` es el parámetro que determina el coeficiente de asociatividad de Hill para las reacciones entre PHOQ y PHOP-P (PHOP fosforilado). Esta nomenclatura se sumariza en el siguiente cuadro, en el cual se indican los rangos de valores considerados válidos desde un punto de vista biológico en el modelado de las redes mediante ODEs.

Prefijo	**Descripción**	**Rango de valores utilizados**
`K`	Coeficiente de activación medio-maximal. Equivale a la concentración en la cual la reacción presenta la mitad de la actividad máxima.	0,001 – 1
`H`	Vida media (inversa de la tasa de degradación).	1 - 100
`nu`	Coeficiente de Hill Valores cercanos a 1 producen curvas casi lineales, incrementándose la forma de s (sigmoidea) al aumentar este parámetro.	1 - 10
`alpha`	Coeficiente de saturación para un reforzador de la transcripción (enhancer.)	0,1 - 10
`r`	Tasa de transformación Por ejemplo para la reacción de transformación de PHOQ a PHOQ-ACT(PhoQ Activado)	0,001 – 10
`P`	Tasa de fosforilación	0,001 – 10
T_0	Constante de tiempo característica Derivada del reemplazo utilizado para llevar el	

	sistema de ecuaciones a una forma no dimensional. Relaciona al tiempo dimensional (t) con el tiempo adimensional (τ): $t = T0\tau$.	

Cuadro 4.1: Explicación del significado de los parámetros utilizados en las ecuaciones diferenciales.

En resumen, el modelado de las redes genéticas mediante ODEs utilizará, por un lado, la concentración de las especies (ARN, Proteínas) en un momento dado y, por el otro, ciertos parámetros cuyos valores se definirán al momento de ejecutar una simulación de la red.

2.1.1. Ejemplo de interacción: Retroalimentación Positiva de un Gen

Esta interacción sucede cuando una proteína X se une al ADN activando o incrementando la trascripción del gen que codifica para la proteína X, con el aumento del ARN_m y la subsecuente producción de proteína X.

Este es uno de los módulos más sencillos que se pueden encontrar en una arquitectura con las características que utilizamos para implementar el método. A continuación se enumeran las reglas que definen a la retroalimentación positiva para el caso de la proteína PHOP, componente del sistema utilizado en el presente trabajo:

- La proteína PHOP se asocia al ADN del gen phop, aumentando la trascripción del ARN phop
- El ARN phop es traducido a la proteína PHOP
- El ARN phop sufre decaimiento espontáneo[5]
- La proteína PHOP sufre decaimiento espontáneo

En la siguiente figura (Figura 4.2) se esquematiza el mecanismo biológico que determina las reglas mencionadas anteriormente, el cual es modelado a una arquitectura, según la librería utilizada de la forma en la que se muestra en la Figura 4.3.

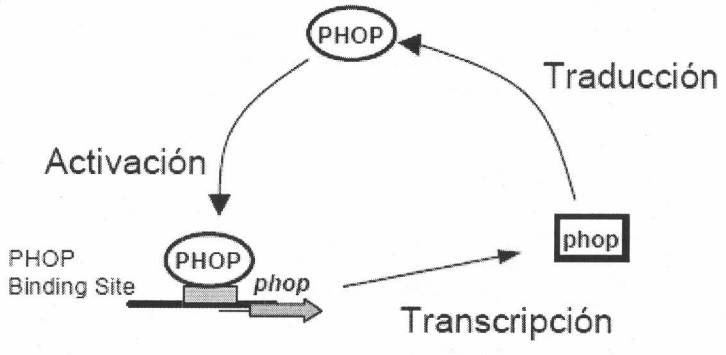


Figura 4.2: Esquema de funcionamiento biológico para la autorregulación positiva de la trascripción del gen phop.
Con el fin de mantener la simplicidad del ejemplo, se muestra la opción más sencilla para la autorregulación positiva de PHOP.

[5] Este decaimiento, que denominaremos decaimiento de primer orden, sucede espontáneamente in vivo para todas las especies

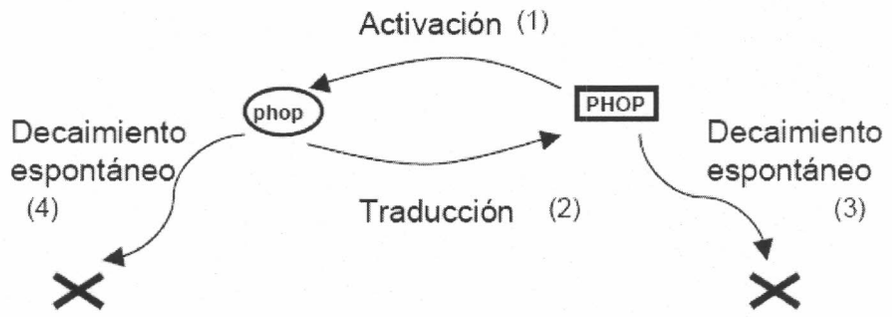


Figura 4.3: Arquitectura del módulo de autorregulación positiva.
Los números en los ejes referencian a los términos de las ecuaciones diferenciales derivadas.

Cada una de las reglas modeladas en la arquitectura de la Figura 4.3, es transformada a una ecuación diferencial que la describe, según la librería de ecuaciones utilizadas en este trabajo. Luego se especifica una ecuación general para cada especie involucrada, componente de la interacción en base a cada una de estas ecuaciones en forma de adiciones o sustracciones, como se ve en la figura 4.4.

$$\frac{d[phop]}{dt} = \frac{T_0}{H_{phop}} \frac{[PHOP]^{\nu_{PHOP\_phop}}}{K_{PHOP\_phop}^{\nu_{PHOP\_phop}} + [PHOP]^{\nu_{PHOP\_phop}}}^{(1)} - \frac{T_0[phop]}{H_{phop}}^{(4)}$$

$$\frac{d[PHOP]}{dt} = \frac{T_0}{H_{PHOP}}[phop]^{(2)} - \frac{T_0}{H_{PHOP}}[PHOP]^{(3)}$$

Figura 4.4: Ecuaciones diferenciales para el módulo de autorregulación positiva.
La numeración entre paréntesis utilizada para cada término se corresponde con la de la reacción del diagrama de arquitectura de la cual se deriva (Figura 4.3).

Como se ve en la figura 4.4 para el cálculo de la concentración del ARN phop, se tienen dos operandos, el primer de ellos representa la tasa de trascripción del gen estimulado por la unión de PHOP al DNA y el segundo término corresponde al decaimiento espontáneo del ARN phop. Para el caso de la concentración de la proteína PHOP, el primero de los términos corresponde a la tasa de traducción del ARN phop a proteína PHOP y el segundo al decaimiento espontáneo de dicha proteína.

También podemos ver que en la primera de las ecuaciones de la figura 4.4 aparecen los parámetros H_{phop}, K_{PHOP_phop}, ν_{PHOP_phop} y en la segunda H_{PHOP}. Para un mayor detalle de los mismos ver el cuadro 4.1.

2.1.2. Ejemplo de interacción: Fosforilación-Defosforilación de una Proteína

Un caso un poco más complejo es el de la fosforilación-defosforilación de una proteína, el cual también aparece en las librerías mencionadas. Esta interacción está mediada por un enzima y puede ejemplificarse mediante la proteína PHOP, vista en el ejemplo anterior, la cual pude ser fosforilada o defosforilada mediante la proteína PHOQ, que posee actividad kinasa sobre la especie defosforilada de PHOP y actividad fosfatasa sobre la especie fosforilada (PHOP-P). Además PHOP-P puede sufrir una defosforilación espontánea. Para este ejemplo, las reglas a considerar serían:

- PHOP es fosforilada a PHOP-P por la acción kinasa de PHOQ
- PHOP-P es defosforilada por la acción fosfatasa de PHOQ
- PHOP-P presenta un cierto nivel de defosforilación espontánea
- La proteína PHOP sufre decaimiento espontáneo
- La proteína PHOP-P sufre decaimiento espontáneo

En la figura 4.5 se muestra el esquema de comportamiento biológico para el módulo utilizando el ejemplo para el sistema de dos componentes PhoP/PhoQ. En la figura 4.6 se muestra la arquitectura definida para el modelado y en la figura 4.7 se muestra el sistema de ecuaciones diferenciales obtenido.

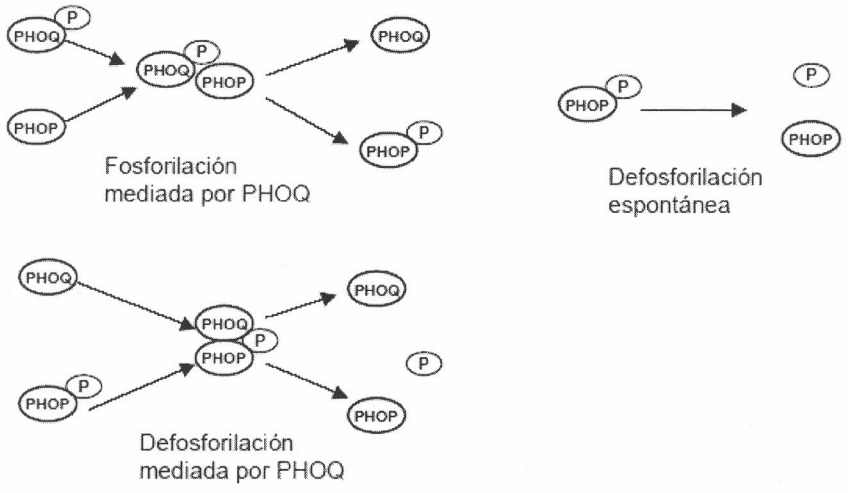


Figura 4.5: Esquema biológico para las reacciones de fosforilación/defosforilación de PHOP.

Para preservar la simplicidad del ejemplo, se modela el esquema más sencillo de fosforilación de PHOP. No se agregan opciones exploradas por el método, como la que distingue dos formas de PHOQ, una normal en medio con alto Mg^{2+} y una activada con bajo Mg^{2+}. Para esta opción, se explorará la combinatoria entre estos esquemas de fosforilación y las enzimas potenciales, PHOQ y PHOQ-ACT (ivado).

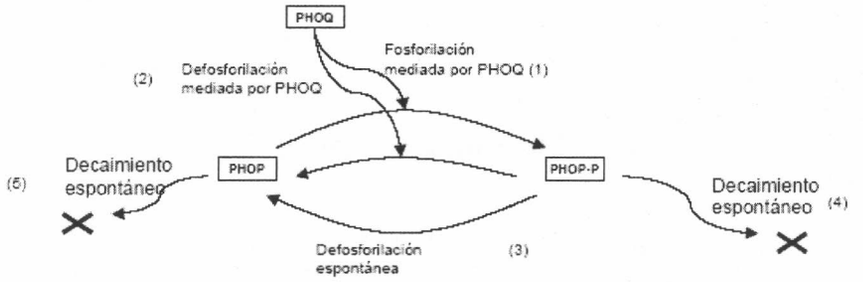


Figura 4.6: Arquitectura del módulo para las reacciones de fosforilación/defosforilación de PHOP mediadas por PHOQ.
Los números en los ejes referencian a los términos de las ecuaciones diferenciales derivadas.

$$\frac{d[PHOP]}{dt} = T_0(P_{PHOQ}[PHOPP]\frac{[PHOQ]^{\nu PHOQ\_PHOPP}}{K^{\nu PHOQ\_PHOPP}_{PHOQ\_PHOPP} + [PHOQ]^{\nu PHOQ\_PHOPP}}^{(2)} + [PHOPP]r^{(3)}_{PHOPP}$$
$$-P_{PHOQ}[PHOP]\frac{[PHOQ]^{\nu PHOQ\_PHOP}}{K^{\nu PHOQ\_PHOP}_{PHOQ\_PHOPP} + [PHOQ]^{\nu PHOQ\_PHOP}}^{(1)} - \frac{[PHOP]}{H_{PHOP}}^{(5)})$$

$$\frac{d[PHOPP]}{dt} = T_0(P_{PHOQ}[PHOP]\frac{[PHOQ]^{\nu PHOQ\_PHOP}}{K^{\nu PHOQ\_PHOP}_{PHOQ\_PHOP} + [PHOQ]^{\nu PHOQ\_PHOP}}^{(1)}$$
$$-P_{PHOQ}[PHOPP]\frac{[PHOPP]^{\nu PHOQ\_PHOPP}}{K^{\nu PHOQ\_PHOPP}_{PHOQ\_PHOPP} + [PHOPP]^{\nu PHOQ\_PHOPP}}^{(2)} - [PHOPP]r^{(3)}_{PHOPP} - \frac{[PHOPP]}{H_{PHOPP}}^{(4)})$$

Figura 4.7: Ecuaciones diferenciales para el módulo de fosforilación – defosforilación.
La numeración entre paréntesis utilizada para cada término se corresponde con la de la reacción del diagrama de arquitectura de la cual se deriva.

Como puede verse en la Figura 4.7, las ecuaciones obtenidas presentan una mayor complejidad, ya que la interacciones que las definen son más complejas desde una perspectiva biológica. Para el caso de la ecuación que determina la concentración de PHOP pueden verse cuatro operandos, siendo la tasa de defosforilación de PHOP-P mediada por PHOQ y la tasa de defosforilación espontánea los términos que suman y la tasa de fosforilación de PHOP mediada por PHOQ y la tasa de decaimiento espontáneo de PHOP los que restan. Mientras que para la concentración de PHOP-P solamente suma la tasa de fosforilación de PHOP mediada por PHOQ, siendo los términos que restan, las tasas de defosforilación de PHOP-P mediada por PHOQ y la espontánea, y la tasa de decaimiento espontáneo de PHOP-P.

Como en la arquitectura que se utilizó en el presente trabajo, las interacciones entre especies vistas en las dos secciones anteriores existen la ecuación que determina la concentración de PHOP deberá tener en cuenta ambas ecuaciones (Figura 4.4 y Figura 4.7), y otras que se verán cuando se estudie la arquitectura de manera completa, más adelante en este capítulo.

2.2. Obtención de soluciones a partir de un Algoritmo Genético

Para poder ejecutar las simulaciones y optimizar la obtención de parámetros se utilizó la herramienta ***Ingeneue*** [16], la cual fue desarrollada específicamente para construir modelos de redes genéticas y explorar su comportamiento mediante simulaciones que exploran patrones temporales y espaciales. Esta herramienta utiliza librerías de módulos biológicos estereotipados para construir los modelos y traducirlos a ODEs, tal cual como necesitamos para nuestro método.

Originalmente, ***Ingeneue*** corre las simulaciones buscando condiciones iniciales de ejecución, esto es, seleccionando valores de parámetros que aseguran el comportamiento del modelo definido en base a condiciones de éxito definidas junto al modelo.

Para poder implementar nuestro método fue necesario modificar esta herramienta para poder incluir un algoritmo genético que optimice el proceso de obtención de soluciones a la arquitectura de entrada en relación a la forma en que lo hace el método basado en búsqueda aleatoria utilizada por esta aplicación. En la primer parte de esta sección explicaremos la forma en la que se ejecutan las simulaciones en esta herramienta a partir de una arquitectura de entrada, para luego explicar la forma en la que se implementó el algoritmo genético dentro de ***Ingeneue***, así como los operadores genéticos utilizados en la optimización.

2.2.1. Simulaciones realizadas sobre los modelos

Los sistemas de ecuaciones diferenciales generados permiten calcular la concentración de cada una de las especies de la red para un instante dado del tiempo en función de las concentraciones de las mismas en el instante anterior. Para realizar la simulación se requiere que el conjunto de parámetros que forman parte del modelo tengan un valor asignado. Este hecho resulta importante dado que los valores de estos parámetros no son, en la mayoría de los casos, conocidos y son justamente, como se explicó antes, las variables que determinan que la ejecución de una simulación sea exitosa o fracase. Al independizarnos de los mismos nuestro método permite realizar predicciones sobre la red genética estudiada sin necesidad de fijarlos arbitrariamente.

Una vez que comienza una simulación se le asigna un valor inicial a todas las especies (nodos de la red) para el tiempo T_0, y luego el motor de ***Ingeneue*** calcula, en base a la integración de las ecuaciones, el valor de cada una de estas especies para los tiempos T_1; T_2; ...; T_n. Si bien, en el presente trabajo se describirán los resultados obtenidos de corridas para red genética PhoP/PhoQ y PmrA/PmrB, estas características aplican para cualquier otra simulación que se realice.

Utilizando un conjunto de valores de concentraciones iniciales definido para cada una de las especies que forman parte de la red es posible calcular las curvas de concentración en función del tiempo para todas las especies involucradas, como se muestra en la figura 4.8.

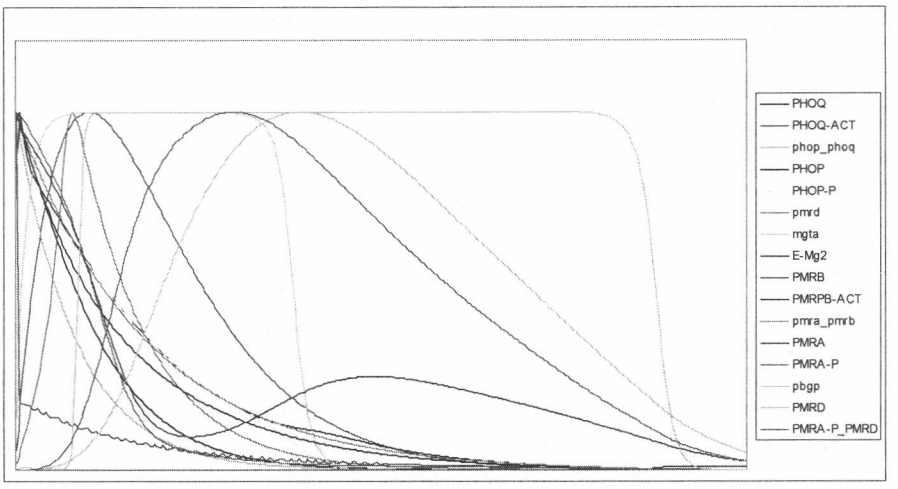


Figura 4.8: Resultados de una corrida de simulación realizada para una arquitectura para un cierto conjunto de parámetros. En la gráfica se muestra la variación de las concentraciones de las especies de la red a lo largo del tiempo de la simulación.

La simulación será considerada "exitosa" (es decir, se ha obtenido una "solución válida" para la red propuesta) si las curvas obtenidas para ciertas especies distinguidas de ARN, en nuestro ejemplo mgta, pbgp y pmrd, son similares a las que se obtendrían al generarse un cierto patrón esperado, como por ejemplo "mgta, pbgp y pmrd con concentración mayor a 0,8 a partir de un cierto instante en la simulación", determinando la activación o no de los genes correspondientes.

Para determinar la habilidad de cierta arquitectura, con su correspondiente conjunto de parámetros para reproducir el comportamiento de los sistemas en el organismo vivo se utilizó una función de *score* basada en los valores de las concentraciones de ciertas especies distinguidas[6] Un score cercano a 0 indica una gran similitud con la limitación impuesta, y un valor cercano a 1 indica una gran diferencia. El *score* se calcula como la combinación de distintas funciones de umbral. Básicamente, se utilizan funciones sigmoideas para testear la expresión de las especies seleccionadas, por ejemplo considerando que una especie se encuentra activada si supera el 10% de su valor maximal de expresión, y apagada en caso contrario. Cada función de umbral utilizada devuelve un valor escalar con el valor medio-maximal en un umbral de acuerdo a las fórmulas:

[6] En nuestro modelo de estudio son las especies mgta, pbgp y pmrd (ARNs), aunque el esquema es válido para cualquier especie genérica. Las dos primeras indican activación de los sistemas PhoP/PhoQ y PmrA/PmrB, respectivamente, y la tercera indica si el sistema PhoP/PhoQ se encuentra ejerciendo un estímulo sobre el sistema PmrA/PmrB.

$$T_{off} = \alpha_{max}\frac{(x_i/x_t)^3}{1+(x_i/x_t)^3} \qquad (4.1)$$
$$T_{on} = \alpha_{max}(1-\frac{(x_i/x_t)^3}{1+(x_i/x_t)^3})$$

donde x_t representa el umbral para la especie x, α_{max} es el peor valor posible (0,5 en este ejemplo), y el subíndice i representa la especie i. La razón para utilizar sigmoideas es que estas funciones responden linealmente cerca del umbral pero proveen tanto penalizaciones como premios decrecientes al alejarse del umbral. Este hecho resulta coherente con la intuición de que si un gen particular se encuentra muy por encima de su umbral de activación, no debería importar cuánto lo está. Las funciones de score individuales son combinadas mediante la siguiente fórmula, que también se satura a altos (malos) valores:

$$Score = \left[\frac{\sum_{i=1}^{\#x} T(x_i)}{1+\sum_{i=1}^{\#x} T(x_i)}\right] \qquad (4.2)$$

2.3. El algoritmo genético

Para la implementación del algoritmo genético se tuvieron en cuenta dos aspectos esenciales; el primero de ellos es en qué punto debería ser llamado el algoritmo y de qué manera, mientras que el segundo punto tiene que ver con analizar la mejor forma de calcular los distintos operadores del algoritmo, en función a las características en la ejecución de las simulaciones por parte de ***Ingeneue***. En el siguiente apartado veremos como se incorporó el algoritmo genético dentro del software, para luego dar una explicación detallada de todos los pormenores que se tuvieron en cuenta para desarrollar el algoritmo genético.

2.3.1. Inclusión del algoritmo genético dentro de Ingeneue

Para poder implementar el algoritmo genético, fue necesario modificar algunas de las clases de ***Ingeneue***, ya que este software no tiene una interfaz que permita conectar cualquier método de obtención de resultados.

Las clases que fueron modificadas fueron las que corresponden al *iterador* y a los *stoppers*. El primero de ellos es quien regula originalmente la ejecución sucesiva de simulaciones a partir de diversas configuraciones. Para nuestro caso esta funcionalidad fue eliminada, ya que el algoritmo genético se encarga de controlar las simulaciones.

La otra clase corresponde a las funciones que van ejecutando la simulación y que devuelven al finalizar el *score* obtenido, como se explicó en el apartado anterior. La modificación de esta consiste en cambiar el método que devuelve el *score* final y suplantarlo por la función de aptitud (Ver más adelante).

A continuación se muestra la interacción entre los componentes de ***Ingeneue*** y el algoritmo genético incorporado:

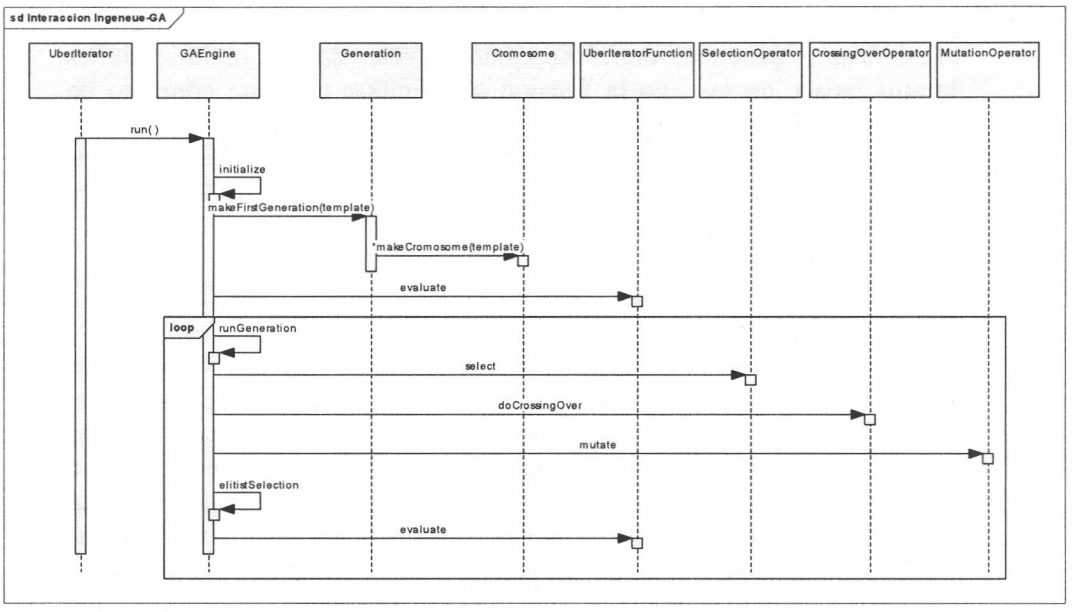


Figura 4.9. Diagrama de secuencia de una ejecución del algoritmo genético dentro de Ingeneue. Los objetos en amarillo pertenecen a Ingeneue

2.3.2. Detalles de implementación del algoritmo genético

Como se explicó anteriormente, ***Ingeneue*** para realizar las corridas de las simulaciones requiere asignar valores a un conjunto de parámetros, los cuales determinan el resultado final que se obtendrá en la ejecución de la red. Una vez que comienza ésta, se van tomando puntos de corte que simulan el paso del tiempo y a partir de allí se van obteniendo las concentraciones de los distintos componentes de la red genética (nodos) y del resultado (*score*) que se va obteniendo en la corrida. Una vez que se llega a un tiempo determinado la simulación finaliza. En nuestro modelo se escogió un tiempo igual a 300, que es equivalente al tiempo que ha sido utilizado en las simulaciones corridas por el método basado en búsqueda aleatoria, con el fin de mantener los mismos valores estándares para la comparación de ambos métodos. De todas maneras este tiempo está medido en unidades genéricas por lo que se puede transpolar a otras unidades.

Este manejo de los puntos de cortes y obtención del resultado de la simulación es realizada en nuestro modelo a partir de nueve funciones llamadas *stoppers* que emiten cada una un valor o *score* al finalizar la ejecución, es decir, al llegar al tiempo de parada establecido. Para calcular el *score* total de la ejecución, se toma el máximo valor entre estos *stoppers*. Luego, este valor deberá ser menor al *threshold* de 0,3 para establecer que la configuración de parámetros tomada al inicio de la simulación da resultados positivos en la corrida.

Esto nos muestra, que el problema que estamos intentando optimizar con el algoritmo genético, tiene las características de las optimizaciones multiobjetivo, en las que la función a optimizar tiene un conjunto de limitaciones para determinar si una solución es factible o no. En nuestro caso, una solución es factible si y solo si el *score* final de la corrida es menor al *threshold*, lo que equivale a decir que cada uno de los *stoppers* devuelva un *score* menor a dicho valor.

El cromosoma

El cromosoma se representó como el conjunto de los valores que tomaban los parámetros para la solución dada. Es decir, representan el estado inicial de la red que será corrida en la simulación. Para el caso de la red que representan a los sistemas PhoP/PhoQ y PmrA/PmrB, son 66 parámetros, por lo tanto, cada cromosoma estará representado por estos 66 genes con valores reales según la especificación de los valores que puedan tomar cada uno de los parámetros, es decir, respetando el rango de valores que puede tomar cada uno, manteniendo su significación biológica.

El diseño del cromosoma es el que se muestra en la siguiente figura:

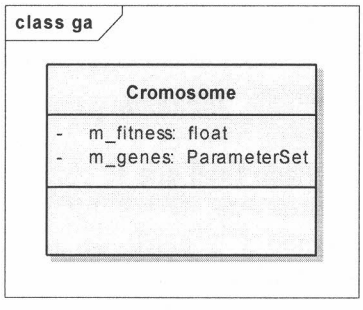


Figura 4.10. Representación del cromosoma

Según se ve en la figura de arriba, el cromosoma guarda la información del conjunto de parámetros iniciales de una ejecución dada. La clase **ParameterSet** pertenece a ***Ingeneue*** y es como éste representa el conjunto de valores de los parámetros, mientras que el atributo **m_fitness** almacena el valor de aptitud de ese individuo.

Función de aptitud

Para la obtención de una función de aptitud útil para el desarrollo del algoritmo genético, además de la naturaleza multiobjetivo del problema a resolver, se suma la baja probabilidad de encontrar resultados exitosos en las primeras generaciones de la ejecución, esto hace que no se pueda utilizar el *score* final devuelto por ***Ingeneue*** luego de las corridas de simulaciones, como valor de aptitud para los individuos. Debido a esto se utilizó un método de *suma pesada* para que la evaluación de un individuo tomara cada uno de los *scores* obtenidos por cada *stopper* y de esa manera tener una forma de comparar dos soluciones que no fueran exitosas, de tal manera de poder saber cual de ellas elegir.

Fue por todas estas razones que se decidió implementar una función que otorgue un castigo (valor positivo) a los *scores* que fueran mayores al *threshold*, y un valor negativo a aquéllos que eran menores a éste.

$$fitness = \sum_{i=1}^{9} (score_i - threshold) * f(score_i) \quad (4.3)$$

siendo,

$$f(x) = \begin{cases} 1 & si\ x < threshold \\ N & si\ x \geq threshold \end{cases}$$

Como puede verse, esta función toma la desviación que hay entre el valor de cada *stopper* ($score_i$) y el *threshold*, castigando a aquellos que se encuentran por encima de este último con un entero positivo (N) lo suficientemente grande como para que, además no solo se vea la distancia entre el valor del *stopper* y el *threshold* sino que además se tenga una idea de cuantos *stoppers* no están cumpliendo con la condición deseada.

Un problema que sucedía con este método era que no todos los individuos con mejor valor de aptitud, obtenían un menor *score* en la ejecución, y debido a que lo que se busca es minimizar este último, al aplicar el operador elitista, el cual se hacía sobre el valor de aptitud, a la siguiente generación pasaban individuos que no eran los mejores para nuestro problema. Debido a ello se modificó la función de aptitud de la siguiente manera:

$$fitness = score_{ingeneue} * \left(\sum_{i=1}^{9} 10^{N+1} * f(score_i) \right) \quad (4.4)$$

siendo,

$$f(x) = \begin{cases} 1 & si\ x > threshold \\ x/10 & si\ x \leq threshold \end{cases}$$

Con N un valor que especifica cuantos decimales del score nos interesa comparar entre las soluciones, $N+1$ nos permite ver cuantos de los *stoppers* superan al *threshold* y al sumar $score_{ingeneue}$ al valor de aptitud nos permite tener en los decimales el *score* original provisto por ***Ingeneue*** que da la corrida de la simulación. De esta manera con un solo valor devuelto tenemos suficiente información para decidir cuales son los mejores individuos de la población y además saber cual será su *score* al ejecutar la simulación. Recordemos que por la definición del problema cuanto menor sea el *score* mejor será la solución obtenida.

Selección

La selección se hizo por torneo en la cual cada individuo de la población compite en 2 oportunidades para ver quién es el que tiene un menor valor de aptitud.

Operador de Cross Over

Para la selección de este operador se hicieron varias pruebas entre las que figuran en la literatura [13]. En un principio se tomo el conjunto de parámetros como una cadena, y de la misma manera en que se hace para cromosomas binarios, se hacía un corte en una posición al azar en ambos padres y se intercambiaban los fragmentos obtenidos. Es decir, si:

$[p_1, p_2, \ldots, p_n]$ y

$[m_1, m_2, \ldots, m_n]$

son las cadenas de parámetros de los padres, entonces se elige un i, $1 \leq i \leq n$ y se formaban los dos hijos (offsprings), de la siguiente forma:

$[p_1, \ldots, p_{i-1}, m_i, \ldots, m_n]$ y
$[m_1, \ldots, m_{i-1}, p_i, \ldots, p_n]$

También se probó realizando dos cortes en la cadena a partir de posiciones $0 \leq i < j \leq n$ e intercambiando de la siguiente manera:

$[p_1, \ldots, p_{i-1}, m_i, \ldots, m_{j-1}, p_j, \ldots, p_n]$ y
$[m_1, \ldots, m_{i-1}, p_i, \ldots, p_{j-1}, m_j, \ldots, m_n]$

Ambos métodos devolvieron resultados similares en las soluciones obtenidas.

Una desventaja importante en este método era que los valores de cada parámetro no se modificaban a la lo largo de las generaciones sino que aparecían relacionados con otros parámetros en diferentes genomas. Es por ello que se agregó un método para obtener nuevos valores de los parámetros dejando de ver al cromosoma como una cadena, sino tomando cada parámetro por separado y aplicando la siguiente función:

$$p_{new} = Bp_i + (1 - B)m_i \qquad (4.5)$$

Siendo, B un número aleatorio entre 0 y 1 y p_i, m_i el i-ésimo parámetro de uno y otro padre, respectivamente.

Finalmente se optó por un esquema que combine estas opciones y para cada parámetro particular se tomaba la decisión de que tome el valor de uno u otro padre, o que se le aplique la función anterior.

Mutación

Para aplicar el operador de mutación en cada simulación se determina antes del comienzo de la ejecución un porcentaje o probabilidad de mutación, de tal manera que para efectuarla se toman todos los genes de todos los individuos de la generación a la que se le está aplicando el operador y se seleccionan a aquellos genes que serán mutados en función al porcentaje determinado. En distintas pruebas se tomaron diferentes proporciones para aplicar este operador, las cuales variaron entre el 1% al 15%.

Criterio de Parada y Tamaño de la Población

El tamaño de la población debe permitir obtener una importante diversidad de soluciones, es por ello que se tomaron diversos valores para analizar el comportamiento del algoritmo.

En cuanto al criterio de parada, se determino que el algoritmo corriera una cantidad fija de generaciones. La decisión de esta estrategia está basada en que no se puede asegurar la convergencia de resultados, ya que por la naturaleza del problema que se está tratando, podría suceder que una generación posea peores resultados que su generación madre. En la sección de resultados se verá un ejemplo de esta situación y además se analizará como fue el comportamiento del algoritmo para corridas de 100 y 250 generaciones.

Política de Elitismo

El elitismo se implementó tomando a los n individuos de la población con mejor valor de aptitud y reemplazándolos por los n individuos con valor de aptitud más bajo. Este porcentaje de elitismo es otro de los parámetros que maneja el algoritmo.

2.4. Análisis de robustez de las soluciones obtenidas

Al igual que se explicó en el método basado en búsqueda aleatoria en el capítulo anterior (sección 2.6), se realiza un análisis de robustez de los parámetros tomando un conjunto de soluciones que demuestren cierto sentido biológico en el comportamiento de sus genes (lo que se hace a partir de las curvas de expresión de cada especie) y se van calculando los valores obtenidos por la red al modificar cada uno de los parámetros (de a uno por vez) a lo largo del conjunto de valores en donde dicho parámetro es biológicamente significativo. (ver figura 3.4 en el capitulo anterior para ver el algoritmo).

3. Resultados

3.1. Introducción

En esta sección presentaremos la ejecución de nuestro método a una red genética constituida por la interacción de los sistemas de dos componentes PhoP/PhoQ y PmrA/PmrB, presentes en *Salmonella enterica serovar Typhimirium*. El sistema PmrA/PmrB (regulador de

respuesta/sensor, respectivamente) responde independientemente a dos señales distintas, por un lado, al alto nivel de Fe^{3+} extracelular, el cual es sensado por PmrB, y por el otro, al bajo nivel de Mg^{2+}, sensado por la proteína PhoQ.

El sistema PhoP/PhoQ (regulador de respuesta/sensor, respectivamente) constituye un regulador maestro que gobierna la adaptación a medios de bajo Mg^{2+} y la virulencia en ratones, así como también otras funciones biológicas. Su funcionamiento como sistema de dos componentes es en principio análogo al anterior, siendo el estímulo activador este bajo nivel de Mg^{2+} extracelular. Entre los genes activados por PhoP se encuentra el gen pmrD, el cual resulta de especial interés porque presenta un sitio de vinculación para PmrA, y su producto, la proteína PmrD, puede asociarse a la proteína PmrA. Ambos sistemas de dos componentes muestran una coordinación de sus funciones in vivo, aunque los mecanismos exactos de interacción son aún desconocidos.

En la sección 3.2 se explica cómo la evidencia de sitos de vinculación para los genes de la red genética estudiada, permite determinar las principales reglas que determinan la interacción entre las especies modeladas. Luego, en la sección 3.3 se explicará el método propuesto basado en algoritmos genéticos. En la sección 3.4 se realizará un análisis de robustez. Dejando para las dos últimas secciones la comparación de resultados con el método basado en búsqueda aleatoria y la comparación entre los resultados de nuestro método y las experimentaciones biológicas, respectivamente.

3.2. Modelado de Interacciones Genéticas Basado en sitios de vinculación

Con el objetivo de determinar modelos para los diferentes sitios de vinculación de factores de transcripción se agruparon ejemplos de la base de datos RegulonDB, los cuales fueron prototipados utilizando modelos de matrices pesadas.

Luego, se realizó una búsqueda en las regiones intergénicas de Salmonella utilizando estos modelos, detectándose la ocurrencia de distintos motivos de sitios de vinculación para factores de transcripción y cajas PhoP (*PhoP boxes*) putativas dentro de las mismas regiones intergénicas. La distancia entre la caja PhoP y los otros sitios de vinculación putativos fueron agrupadas. Las mencionadas distancias fueron prototipadas utilizando funciones de agrupamiento difuso (*fuzzy membership*) y utilizadas para caracterizar la relación entre las cajas PhoP putativas y otras cajas de vinculación (*binding-boxes*) mediante la evaluación de su distancia de acuerdo con casos previamente conocidos reportados en bases de datos. Se utilizó la herramienta SOAR [28] para identificar interacciones de regulación involucrando a la proteína PhoP. En la figura 4.11 se observan ejemplos de distintas regiones promotoras encontradas. En la figura 4.12 se muestra el esquema de interacción entre los genes de los subsistemas PhoP/PhoQ y PmrA/PmrB derivado de la información obtenida del estudio de los sitios de vinculación.

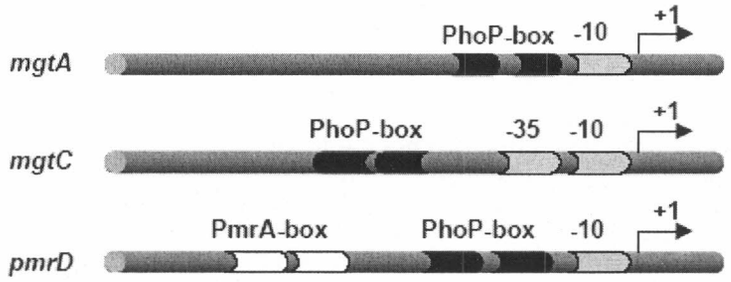


Figura 4.11: Regiones promotoras de los genes mgtA, mgtC y PmrD.

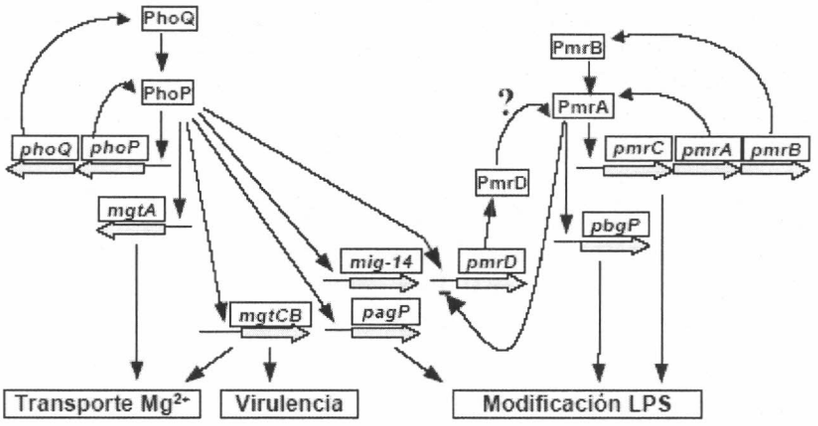


Figura 4.12: Interacciones entre los genes de los subsistemas PhoP/PhoQ y PmrA/PmrB derivados de la información obtenida del estudio de los sitios de vinculación.

3.2.1. Descripción del modelo utilizado

En base a la combinación de la evidencia acumulada mediante el estudio de los sitios de vinculación de los genes de Salmonella, explicados en la sección anterior, y de las características básicas de estos sistemas presentes en la literatura, se utilizó un modelo de la red genética para los sistemas PmrA/PmrB y PhoP/PhoQ que describiremos a continuación

Las reglas que rigen este modelo son las siguientes:

1. El sistema PhoP/PhoQ es activado en bajo nivel de Mg^{2+} extracelular
2. El sistema PmrA/PmrB es activado en alto nivel de Fe^{3+} extracelular.
3. Estas condiciones son censadas por las proteínas de membrana PhoQ y PmrB, respectivamente, las cuales pasan en presencia de la señal a un estado "activado" en el cual las especies actúan como kinasas y fosfatasas de sus reguladores de respuesta asociados, por ejemplo, PhoP y PmrA, respectivamente.
4. Las formas fosfatadas de estas últimas dos especies funcionan como activadoras de la transcripción de distintos genes, tomando en nuestro ejemplo a mgtA para PhoP y pbgp para PmrA.

5. Las proteínas PhoP y PhoQ son traducidas de un mismo ARN phop_phoq, constituyendo un operón.
6. Análogamente, PmrA y PmrB son traducidas de un mismo ARN pmra_pmrb. Estos dos operones son regulados positivamente por PhoP y PmrA, respectivamente.

Con respecto a la interconexión entre ambos sistemas de dos componentes, consideraremos la existencia de una conexión de "ida" desde el subsistema PhoP/PhoQ al subsistema PmrA/PmrB, y una conexión de "vuelta" en sentido inverso. La evidencia de sitios de vinculación y de experimentos de CHIP permite afirmar que la expresión de pmrD es inhibida por PmrA, siendo esta la conexión de vuelta entre los sistemas. Dado que la evidencia de sitios de vinculación permite afirmar que no existe una interacción entre el producto del gen pmrD y el DNA del gen PmrA, que constituiría una conexión de ida entre los sistemas a nivel transcripcional, se asume que la interacción entre pmrD y PmrA es post-transcripcional. Esta interacción consiste en que la forma fosfatada de PmrA es protegida por PmrD de la actividad fosfatasa de PmrB.

En la figura. 4.13 puede apreciarse un esquema del modelo utilizado. Con respecto a las convenciones de nombres utilizadas para las especies, pueden darse las siguientes definiciones: se utilizan letras mayúsculas para las especies proteicas, minúsculas para los ARN, el sufijo -P para las especies fosforiladas y el sufijo -ACT para identificar especies proteicas que sufren un cambio que modifica su actividad. Las especies que son ácidos ribonucleicos (ARN), son representadas con círculos, mientras que las especies proteicas son representadas con rectángulos.

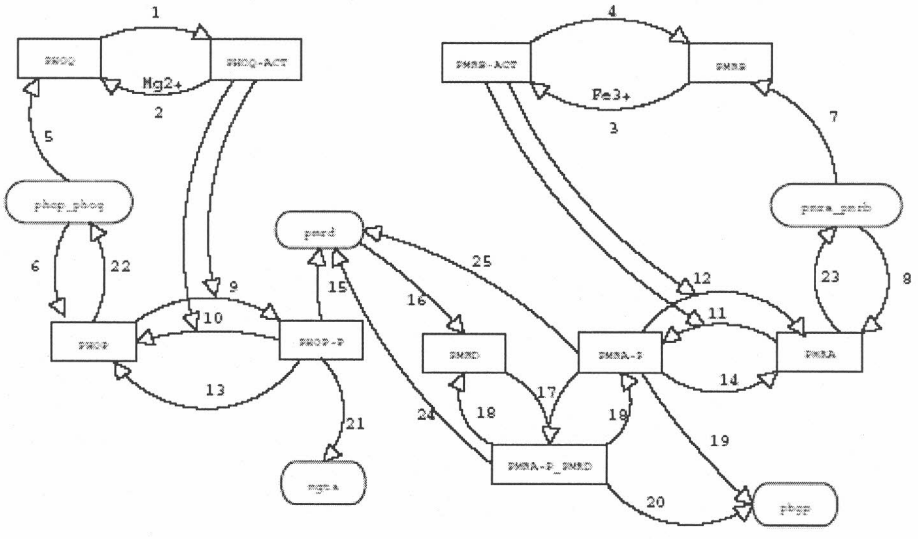


Figura 4.13: Modelo de los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ.

1 Bajo nivel de Mg^{2+} favorece el estado PHOQ ACT (IVADO) en el equilibrio
2 Alto nivel de Mg^{2+} favorece el estado PHOQ en el equilibrio
3 Alto nivel de Fe^{3+} favorece el estado PMRB ACT (IVADO) en el equilibrio
4 Bajo nivel de Fe^{3+} favorece el estado PMRB en el equilibrio
5 phop_phoq es traducido a PHOQ

6 phop_phoq es traducido a PHOP
7 pmra_pmrb es traducido a PMRB
8 pmra_pmrb es traducido a PMRA
9 PHOP es fosforilado a PHOP-P via PHOQ-ACT actividad kinasa
10 PHOP-P es defosforilado a PHOP via PHOQ-ACT actividad fosfatasa
11 PMRA es fosforilado a PMRA-P via PMRB-ACT actividad kinasa
12 PMRA-P es defosforilado a PMRA vía PMRB-ACT actividad fosfatasa
13 PHOP-P es defosforilado a PHOP vía defosforilación espontánea
14 PMRA-P es defosforilado a PMRA vía defosforilación espontánea
15 PHOP-P activa la transcripción de pmrd
16 pmrd es traducido a PMRD
17 PMRD se asocia con PMRA-P para constituir la especie PMRA-P PMRD que activa la
expresión de pbgp e inhibe la de pmrd pero no es afectada por la actividad fosfatasa de PMRB ACT
18 PMRA-P PMRD se disocia para formar PMRD y PMRA-P
19 PMRA-P activa la transcripción de pbgp
20 PMRA-P PMRD activa la transcripción de pbgp
21 PHOP-P activa la transcripción de mgta
22 PHOP activa la transcripción de phop_phoq
23 PMRA activa la transcripción de pmra_pmrb
24 PMRA-P_PMRD inhibe la transcripción de pmrd
25 PMRA-P inhibe la transcripción de pmrd

De acuerdo a la bibliografía analizada, puede inferirse un primer patrón de actividad (o "salida" del modelo) para los sistemas en función de distintos estímulos, o "entradas". Estos patrones se muestran en el cuadro 4.2 mediante valores booleanos (1 o 0) por simplicidad[7], debiendo considerarse que un 1 indica activado, y un 0 inactivado. En el caso del Mg^{2+}, un 1 indica la presencia del estímulo (bajo Mg^{2+} en el caso estudiado) y un 0 el caso contrario. En el Fe^{3+}, un 1 implica la presencia de estímulo (es decir alto Fe^{3+} en el sistema estudiado) y un 0 el caso contrario. Los valores de las concentraciones de las especies Fe^{3+} y Mg^{2+} corresponden a la "entrada" de estos sistemas, mientras que los valores de mgta, pbgp, y pmrd corresponden a la "salida" de los mismos (es decir, la señal traducida). Valores altos de mgta y pbgp indican la activación de los subsistemas PhoP/PhoQ y PmrA/PmrB, respectivamente.

Un valor alto de pmrd indica la activación de la conexión de ida entre los sistemas, y un valor bajo indica la activación de la conexión de vuelta (retroalimentación negativa). En general, los valores indicados en las tablas para las especies de entrada se mantienen constantes a lo largo de las simulaciones, mientras que los de las especies de salida son los esperados al transcurrir la simulación.

Limitación	Mg	Fe	mgta	pmrd	pbgp
1	1	1	1	0	1

[7] En todos los experimentos, se utilizaron valores reales para las concentraciones de las especies, y se consideraron distintos umbrales de activación. Sin embargo, se presentan los patrones en función de especies activadas o no activadas para facilitar el análisis conceptual de las funcionalidades.

2	1	0	1	1	0
3	0	1	0	0	1
4	0	0	0	0	0

Cuadro 4.2: Patrones de entrada-salida para el modelo presentado

En el caso de la segunda limitación, con Mg^{2+} activo y Fe^{3+} inactivo, debe notarse que se pide que el gen pbgp (el cual es activado por el sistema PmrA/PmrB; la activación de pbgp modela en forma genérica la activación de este sistema) no se encuentre activado. La información analizada hasta este punto referente a la actividad post-transcripcional de PMRD no permite modelar un mecanismo que activaría pbgp en ausencia de PMRA-P, por lo que la activación de pbgp depende de la presencia del estímulo de Fe^{3+} en este modelo.

Otro punto de interés consiste en que la trascripción de los operones phoP_phoQ y pmrA_pmrB depende de las especies defosforiladas de PhoP y PmrA. Esta es una hipótesis conservadora a la luz de la evidencia de vinculación que no distingue entre una y otra especie. Este punto será analizado nuevamente en secciones subsiguientes.

Ecuaciones diferenciales para el modelo

Los superíndices en los términos de las ecuaciones indican la reacción del cuadro 4.3 y de la figura 4.14 de la cual se derivan.

Especie	Adición	Sustracción
phop_phoq	22	
PHOQ	2, 5	1
PHOQ-ACT	1	2
PHOP	6, 10,13	9
PHOP-P	9	10, 13
mgta	21	
pmra_pmrb	23	
PMRB	4, 7	3
PMRB-ACT	3	4
PMRA	8, 12, 14	11
PMRA-P	11, 18	12, 14, 17
pbgp	19, 20	
PMRD	16, 18	17
PMRA-P_PMRD	17	18
pmrd	15	24, 25

Cuadro 4.3: Términos de las ecuaciones que determinan las concentraciones de las especies.

Adición: Reacciones que determinan un término sustracción en la ecuación diferencial de la especie.

Sustracción: Reacciones que determinan un término de substracción en la ecuación diferencial de la especie..
Decaimiento: A todas las especies se les agrega un término de sustracción extra para modelar el decaimiento de primer orden.
La numeración se corresponde con la utilizada para numerar los ejes en la Figura 3.3.

$$\frac{\mathrm{d}[\mathrm{PHOQ}]}{\mathrm{dt}} = T_0\Big(\frac{1}{H_{PHOQ}}[phop\_phoq]^{(5)} + r_{PHOQACT}max\_Mg2[PHOQACT][Mg2]^{(2)} - [PHOQ]r^{(1)}_{PHOQ2PHOQACT} - \frac{[PHOQ]^{(dec)}}{H_{PHOQ}}\Big)$$

$$\frac{\mathrm{d}[\mathrm{phop\_phoq}]}{\mathrm{dt}} = T_0\Big(\frac{1}{H_{phop\_phoq}}\frac{[PHOP]^{\nu PHOP\_phop\_phoq}}{K^{\nu PHOP\_phop\_phoq}_{PHOP\_phop\_phoq} + [PHOP]^{\nu PHOP\_phop\_phoq}}{}^{(22)} - \frac{[phop\_phoq]^{(dec)}}{H_{phop\_phoq}}\Big)$$

$$\frac{\mathrm{d}[\mathrm{PHOQ\_ACT}]}{\mathrm{dt}} = T_0\Big([PHOQ]r^{(1)}_{PHOQ2PHOQACT} - r_{PHOQACT}max\_EMg2[PHOQACT][EMg2]^{(2)} - \frac{[PHOQACT]^{(dec)}}{H_{PHOQACT}}\Big)$$

$$\begin{aligned}\frac{\mathrm{d}[\mathrm{PHOP}]}{\mathrm{dt}} = T_0\Big(&\frac{1}{H_{PHOP}}[phop\_phoq]^{(6)} \\ &+P_{PHOQACTP}[PHOPP]\frac{[PHOQACT]^{\nu PHOQACT\_PHOPP}}{K^{\nu PHOQACT\_PHOPP}_{PHOQACT\_PHOPP} + [PHOQACT]^{\nu PHOQACT\_PHOPP}}{}^{(10)} \\ &+r_{PHOPP}[PHOPP]^{(13)} \\ &-P_{PHOQACTK}[PHOP]\frac{[PHOQACT]^{\nu PHOQACT\_PHOP}}{K^{\nu PHOQACT\_PHOP}_{PHOQACT\_PHOP} + [PHOQACT]^{\nu PHOQACT\_PHOP}}{}^{(9)} \\ &-\frac{[PHOP]^{(dec)}}{H_{PHOP}}\Big)\end{aligned}$$

$$\begin{aligned}\frac{\mathrm{d}[\mathrm{PHOPP}]}{\mathrm{dt}} = T_0\Big(&P_{PHOQACTK}[PHOP]\frac{[PHOQACT]^{\nu PHOQACT\_PHOP}}{K^{\nu PHOQACT\_PHOP}_{PHOQACT\_PHOP} + [PHOQACT]^{\nu PHOQACT\_PHOP}}{}^{(9)} \\ &-P_{PHOQACTP}[PHOPP]\frac{[PHOQACT]^{\nu PHOQACT\_PHOPP}}{K^{\nu PHOQACT\_PHOPP}_{PHOQACT\_PHOPP} + [PHOQACT]^{\nu PHOQACT\_PHOPP}}{}^{(10)} \\ &-[PHOPP]r^{(13)}_{PHOPP} - \frac{[PHOPP]^{(dec)}}{H_{PHOPP}}\Big)\end{aligned}$$

$$\frac{\mathrm{d}[\mathrm{mgta}]}{\mathrm{dt}} = T_0\Big(\frac{1}{H_{mgta}}\frac{[PHOPP]^{\nu PHOPP\_mgta}}{K^{\nu PHOPP\_mgta}_{PHOPP\_mgta} + [PHOPP]^{\nu PHOPP\_mgta}}{}^{(21)} - \frac{[mgta]^{(dec)}}{H_{mgta}}\Big)$$

$$\frac{\mathrm{d}[\mathrm{pmra\_pmrb}]}{\mathrm{dt}} = T_0\Big(\frac{1}{H_{pmra\_pmrb}}\frac{[PMRAP]^{\nu PMRAP\_pmra\_pmrb}}{K^{\nu PMRAP\_pmra\_pmrb}_{PMRAP\_pmra\_pmrb} + [PMRAP]^{\nu PMRAP\_pmra\_pmrb}}{}^{(23)} - \frac{[pmra\_pmrb]^{(dec)}}{H_{pmra\_pmrb}}\Big)$$

$$\frac{\mathrm{d}[\mathrm{PMRB}]}{\mathrm{dt}} = T_0\Big(\frac{1}{H_{PMRB}}[pmrd]^{(7)} + r_{PMRBACT}[PMRBACT]^{(4)} - r_{PMRBACT}max\_Fe3[PMRBACT][FE3+]^{(3)} - \frac{[PMRB]^{(dec)}}{H_{PMRB}}\Big)$$

$$\frac{\mathrm{d}[\mathrm{PMRBACT}]}{\mathrm{dt}} = T_0\Big(r_{PMRBACT}max\_Fe3 + [PMRB][Fe3+]^{(3)} - [PMRBACT]r^{(4)}_{PMRBACT} - \frac{[PMRB]^{(dec)}}{H_{PMRB}}\Big)$$

Figura 4.14: Ecuaciones diferenciales de la red genética propuesta (Continuación).

$$\frac{d[PMRA]}{dt} = T_0(\frac{1}{H_{PMRA}}[pmra\_pmrb]^{(8)} + r_{PMRAP}[PMRAP]^{(14)}$$
$$+P_{PMRBACTP}[PMRAP]\frac{[PMRBACT]^{\nu_{PMRBACT\_PMRA}}}{K^{\nu_{PMRBACT\_PMRA}}_{PMRBACT\_PMRA} + [PMRBACT]^{\nu_{PMRBACT\_PMRA}}}^{(12)}$$
$$-P_{PMRBACTK}[PMRA]\frac{[PMRBACT]^{\nu_{PMRBACT\_PMRAP}}}{K^{\nu_{PMRBACT\_PMRAP}}_{PMRBACT\_PMRAP} + [PMRBACT]^{\nu_{PMRBACT\_PMRAP}}}^{(11)}$$
$$-\frac{[PMRA]^{(dec)}}{H_{PMRA}})$$

$$\frac{d[PMRAP]}{dt} = T_0(P_{PMRBACTP}[PMRA]\frac{[PMRBACT]^{\nu_{PMRBACT\_PMRA}}}{K^{\nu_{PMRBACT\_PMRA}}_{PMRBACT\_PMRA} + [PMRBACT]^{\nu_{PMRBACT\_PMRA}}}^{(11)}$$
$$+\frac{[PMRAP\_PMRD]^{(18)}}{H_{PMRAP\_PMRD}}$$
$$-P_{PMRBACTK}[PMRAP]\frac{[PMRBACT]^{\nu_{PMRBACT\_PMRAP}}}{K^{\nu_{PMRBACT\_PMRAP}}_{PMRBACT\_PMRAP} + [PMRBACT]^{\nu_{PMRBACT\_PMRAP}}}^{(12)}$$
$$-r_{PMRAP}[PMRAP]^{(14)} - r_{PMRAP\_PMRD}max\_PMRD[PMRAP][PMRD]^{(17)}$$
$$-\frac{[PMRAP]^{(dec)}}{H_{PMRAP}})$$

$$\frac{d[pbgp]}{dt} = T_0(\frac{1}{H_{pbgp}}(1-(1-alpha_{PMRAP}\frac{1}{H_{pbgp}}\frac{[PMRAP]^{\nu_{PMRAP\_pbgp}}}{K^{\nu_{PMRAP\_pbgp}}_{PMRAP\_pbgp} + [PMRAP]^{\nu_{PMRAP\_pbgp}}})$$
$$(1-alpha_{PMRAP\_PMRD}\frac{1}{H_{pbgp}}$$
$$\frac{[PMRAP\_PMRD]^{\nu_{PMRAP\_PMRD\_pbgp}}}{K^{\nu_{PMRAP\_PMRD\_pbgp}}_{PMRAP\_PMRD\_pbgp} + [PMRAP\_PMRD]^{\nu_{PMRAP\_PMRD\_pbgp}}}))^{(19,20)}$$
$$-\frac{[pbgp]^{(dec)}}{H_{pbgp}})$$

$$\frac{d[PMRD]}{dt} = T_0(\frac{1}{H_{PMRD}}[pmrd]^{(16)} - r_{PMRAP\_PMRD}max\_PMRD[PMRAP][PMRD]^{(17)}$$
$$-\frac{[PMRD]^{(dec)}}{H_{PMRD}})$$

$$\frac{d[PMRAP\_PMRD]}{dt} = T_0(r_{PMRAP\_PMRD}max\_PMRD[PMRAP][PMRD]^{(17)}$$
$$-\frac{[PMRAP\_PMRD]^{(18)}}{H_{PMRAP\_PMRD}})$$

$$\frac{d[pmrd]}{dt} = T_0(\frac{1}{H_{pmrd}}\frac{[PHOPP]^{\nu_{PHOPP\_pmrd}}}{K^{\nu_{PHOPP\_pmrd}}_{PHOPP\_pmrd} + [PHOPP]^{\nu_{PHOPP\_pmrd}}}$$
$$(1-\frac{[PMRAP]^{\nu_{PMRAP\_pmrd}}}{K^{\nu_{PMRAP\_pmrd}}_{PMRAP\_pmrd} + [PMRAP]^{\nu_{PMRAP\_pmrd}}})$$
$$(1-\frac{[PMRAP\_PMRD]^{\nu_{PMRAP\_pmrd}}}{K^{\nu_{PMRAP\_pmrd}}_{PMRAP\_pmrd} + [PMRAP\_PMRD]^{\nu_{PMRAP\_pmrd}}})^{(15,24,25)} - \frac{[pmrd]^{(dec)}}{H_{pmrd}})$$

Figura 4.14: Ecuaciones diferenciales de la red genética propuesta (Continuación).

3.3. Obtención de soluciones por el Algoritmo Genético

En esta sección analizaremos los resultados obtenidos por las distintas configuraciones del algoritmo genético utilizado según sus operadores, el tamaño de población y cantidad de generaciones, dejando para la próxima sección el análisis de la robustez de las soluciones obtenidas.

En el siguiente cuadro se muestran los resultados de distintas configuraciones ejecutadas del algoritmo genético, allí se muestra el tipo de operador de Cross Over utilizado, el tamaño de la población, el número de generaciones ejecutadas, el mínimo score de la red genética obtenida durante la corrida de una simulación y la generación en la que se obtuvo la primer solución en promedio. Recordamos que los operadores utilizados fueron:

- Tipo 1: Método de blending
- Tipo 2: Intercambio de genes de los padres
- Tipo 3: Según un valor al azar, cada gen podía ser calculado utilizando el tipo 1 o el tipo 2.

Cross Over	Tamaño Población	# Generaciones	Min. Score	1º Solución
Tipo 1	50	100	0,218965	20,67
Tipo 1	200	100	0,181234	18,67
Tipo 1	50	250	0,099894	24,00
Tipo 2	50	100	0,195964	20,67
Tipo 2	200	100	0,037984	10,00
Tipo 2	50	250	0,036886	11,67
Tipo 3	50	100	0,191488	20,67
Tipo 3	200	100	0,052287	9,00
Tipo 3	50	250	0,047323	22,00

Cuadro 4.4. Ejecuciones del algoritmo genético.

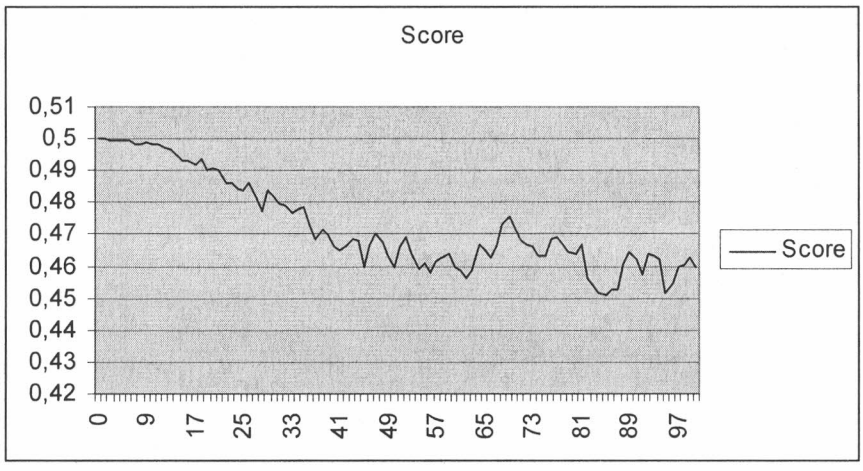


Figura 4.15. Promedio de score por generación para una ejecución del GA

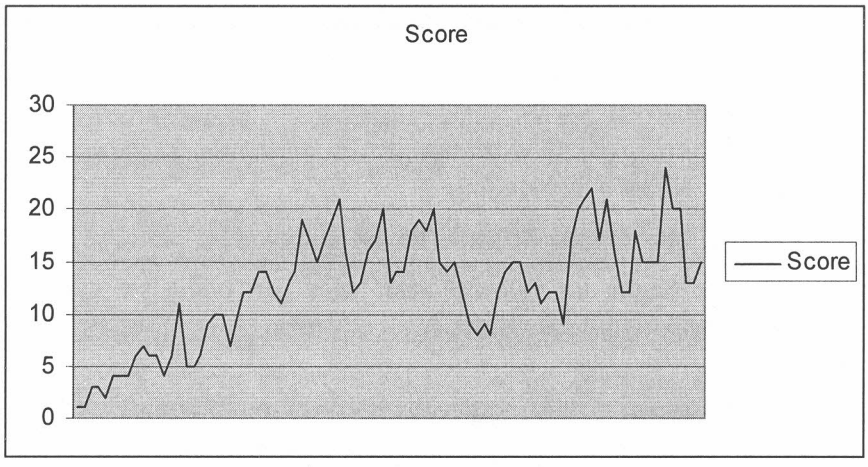


Figura 4.16. Cantidad de soluciones encontradas por generación para una población de 200 individuos y 100 generaciones.

Como se puede deducir la mayor cantidad de de individuos de una población permitió obtener mejores resultados, ya que brindó mayor diversidad en los valores de los parámetros.

3.4. Análisis de robustez de las soluciones obtenidas

En la siguiente figura (4.17) podemos ver como se comportan los genes cuya expresión intentamos estudiar en este trabajo a lo largo de la corrida de una simulación, pudiendo ver que mgta se expresa primero, luego pmrd y por último pbgp.

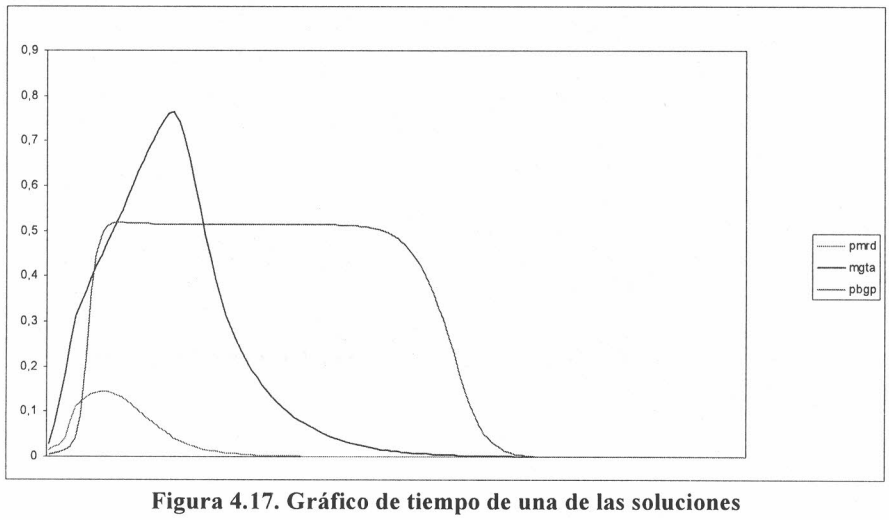


Figura 4.17. Gráfico de tiempo de una de las soluciones

Como primer paso en el análisis de la robustez de las soluciones obtenidas, se utilizó una función iterativa cuya entrada es una solución, es decir una configuración inicial de parámetros, y en cada paso toma uno de ellos, va modificando su valor dentro del rango biológicamente válido (dejando los restantes con el valor de entrada) y ejecuta la red genética para obtener el score. Esta función también permite determinar

la cantidad de puntos en el que se divide el rango biológicamente válido para analizar la robustez de cada parámetro. Para ejemplificar este paso, se puede ver en la figura 4.18. la ejecución de esta función para una solución escogida aleatoriamente dentro de las obtenidas por nuestro método a la red genética de Salmonella, descripta arriba y una división del rango en 100 puntos.

Como se explicó anteriormente, decimos que una configuración de parámetros de la red es una solución, si luego de la ejecución de la simulación de la red genética, se obtiene un score menor a 0,3. En la figura 4.18 puede verse también este valor de *threshold* lo que nos permite determinar en que rangos las soluciones siguen siendo válidas.

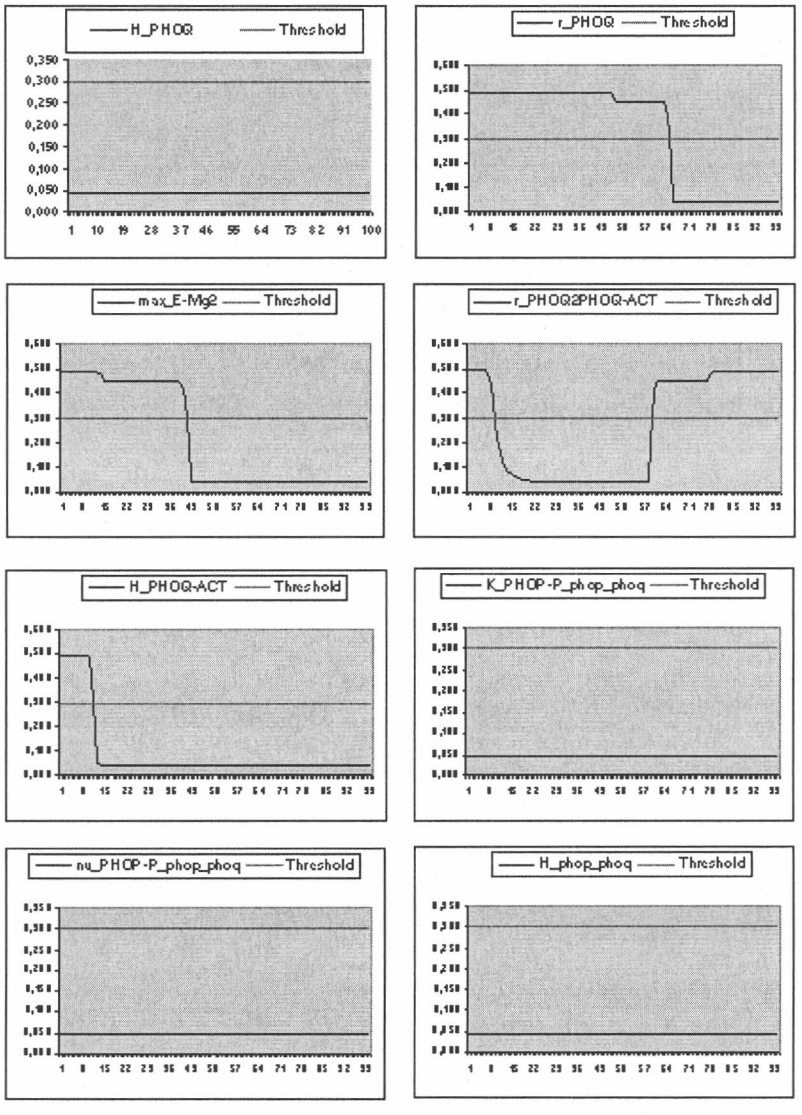


Figura 4.18. Gráfico de robustez de parámetros.

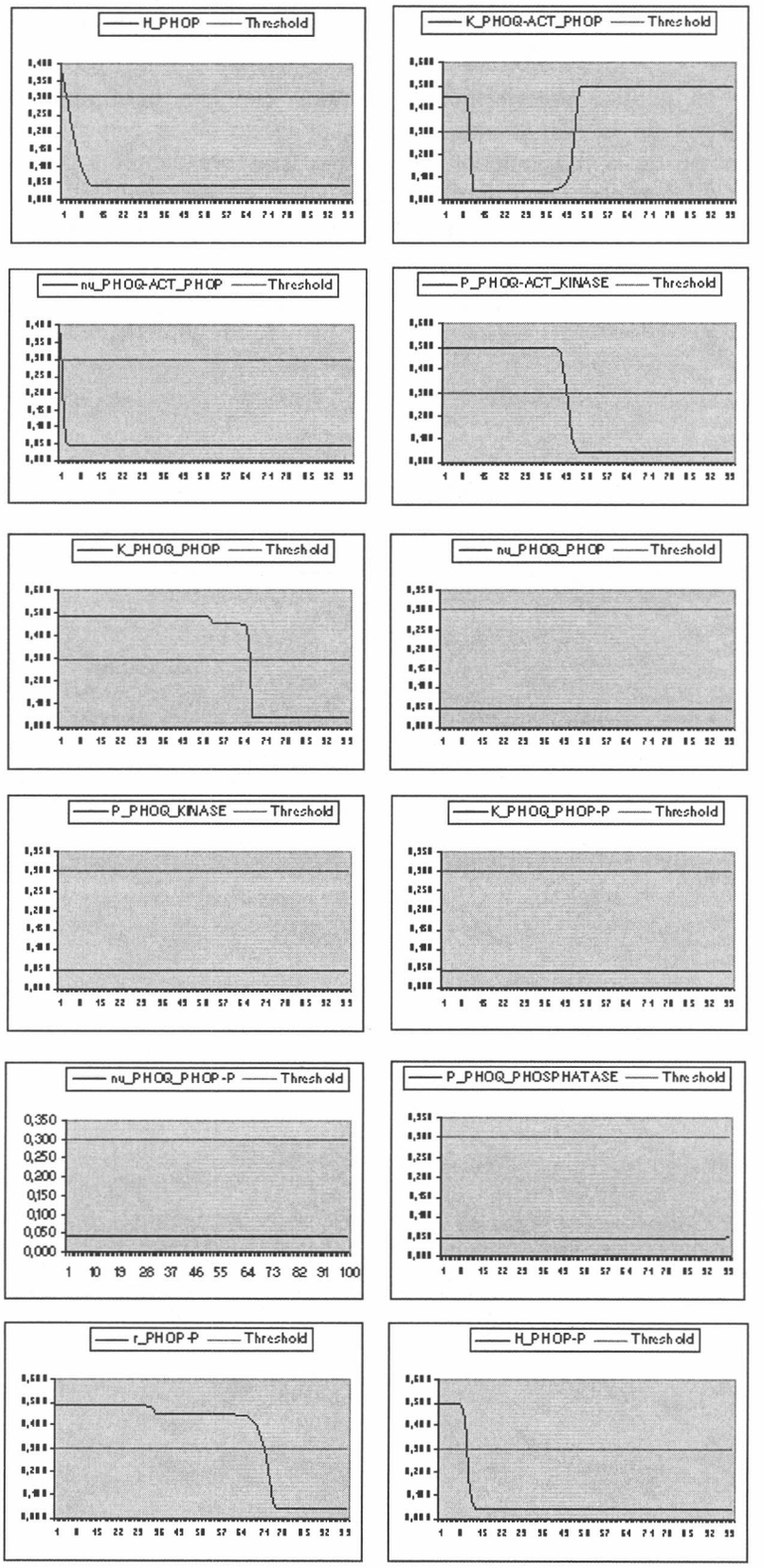


Figura 4.18. Gráfico de robustez de parámetros (Continuación).

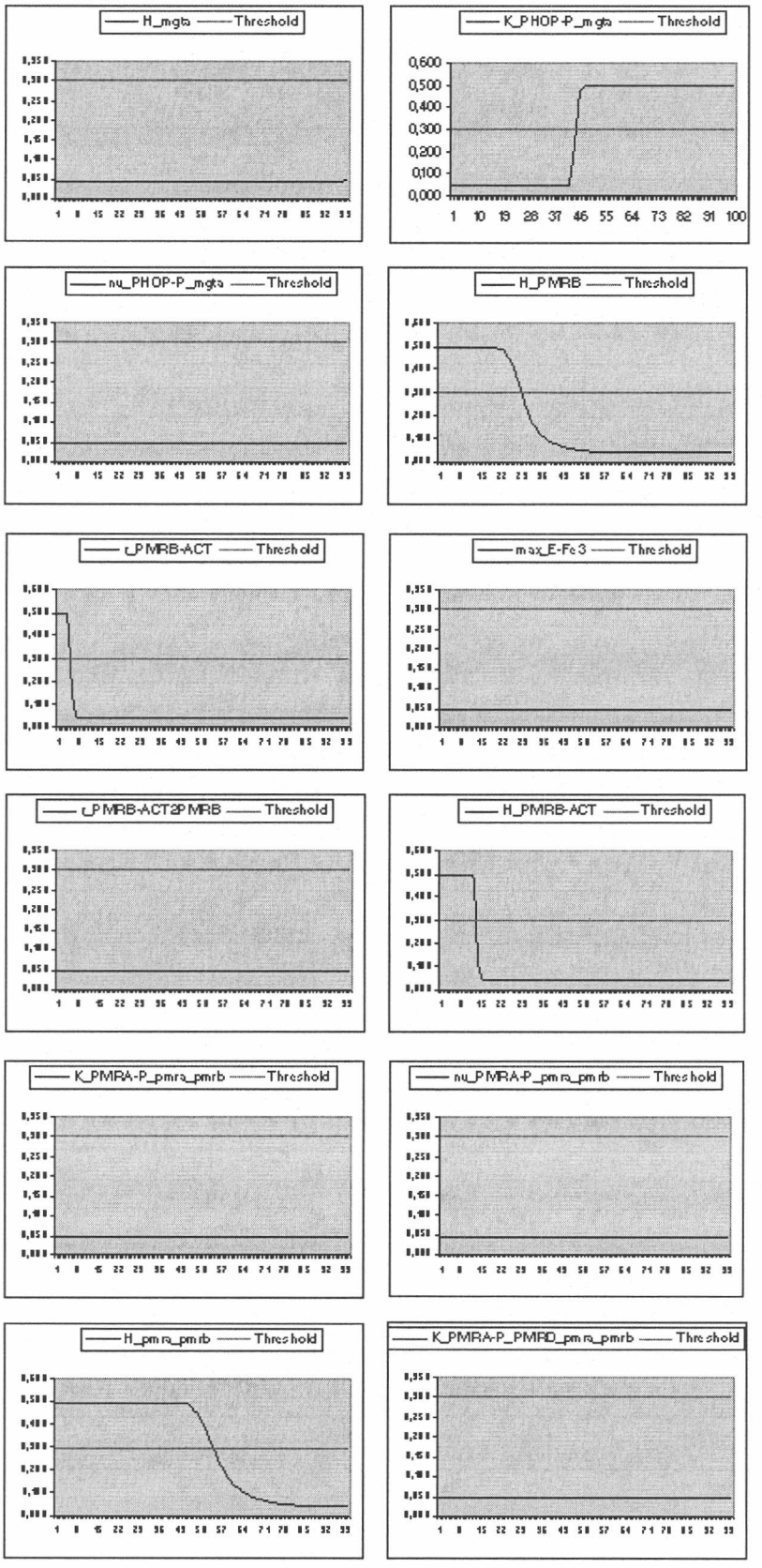


Figura 4.18. Gráfico de robustez de parámetros (Continuación).

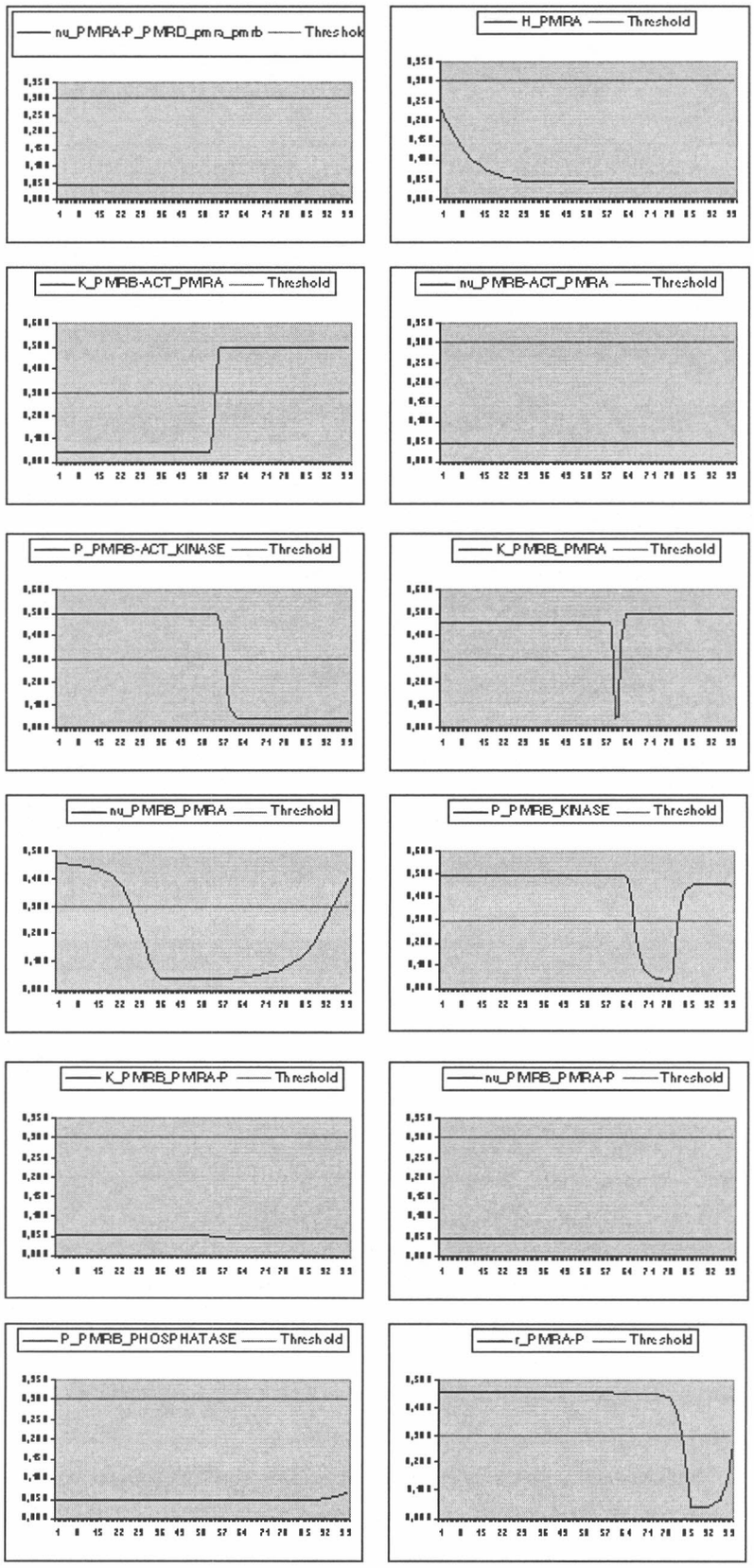


Figura 4.18. Gráfico de robustez de parámetros (Continuación).

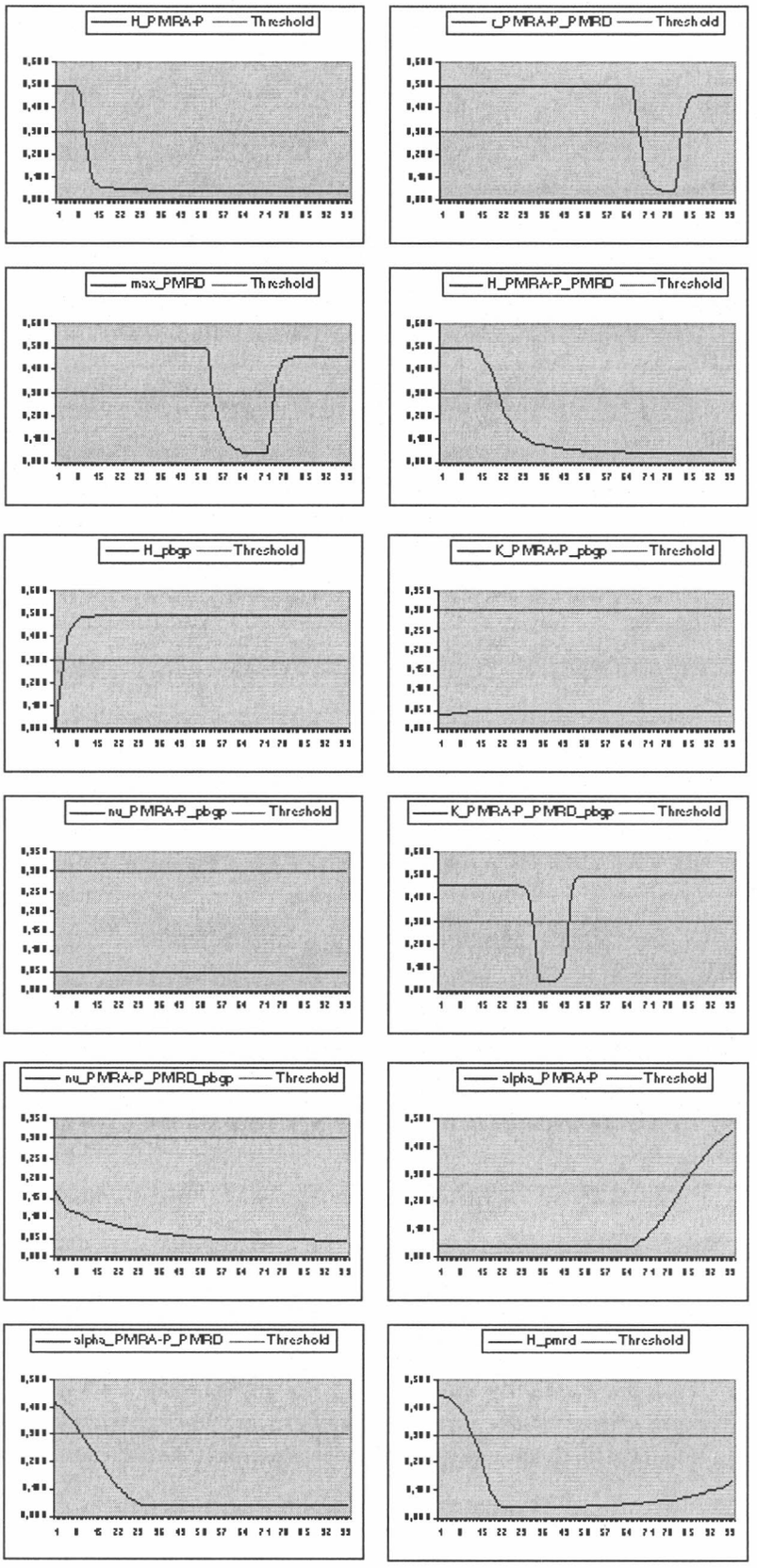


Figura 4.18. Gráfico de robustez de parámetros (Continuación).

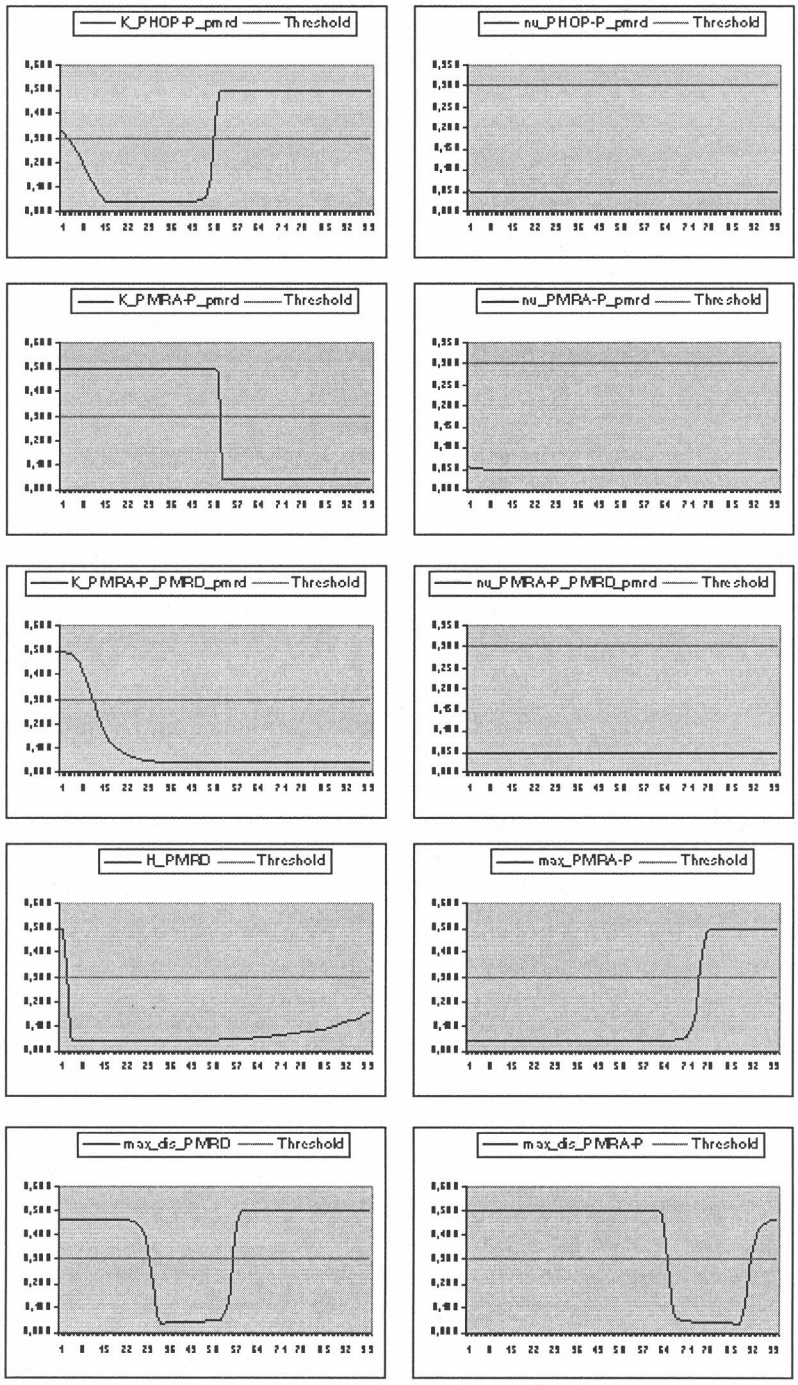


Figura 4.18. Gráfico de robustez de parámetros (Continuación).

En estas figuras puede evidenciarse que algunos parámetros pueden tomar valores con varios órdenes de magnitud de diferencia, incluso algunos de ellos pueden tomar cualquier valor dentro del rango biológicamente válido y así manteniendo el *score* de la red, como por ejemplo $nu_{\_PHOP\_mgta}$. Esto claramente muestra una robustez de la red con respecto a los mismos. Sin embargo, se puede ver que existen otros parámetros, cuya variación afecta al *score* de mayor o menor manera,

mostrando una sensibilidad de la red con respecto a éstos, e incluso provocando que el *score* luego de la ejecución supere el valor del *threshold*, haciendo que la solución deje de ser válida. Esta sensibilidad podría explicar los resultados obtenidos en la ejecución del algoritmo genético, en el sentido de la variabilidad que podía encontrarse entre generaciones sucesivas con respecto a la cantidad de soluciones encontradas. Como ejemplo de este caso podemos observar el gráfico de robustez que muestra K_{PMRB_PMRA}.

Dentro de este último conjunto de parámetros es importante identificar el grado de sensibilidad a los cambios en sus valores, es decir que porcentaje de variación dentro del rango biológicamente significativo puede tomar sin afectar al *score* de manera significativa. Por ejemplo, podemos ver que K_{PMRB_PMRA} provoca una gran variación en el *score* con pequeñas variaciones de su valor, con respecto al obtenido en la simulación, mientras que $K_{PMRA\text{-}P\_PMRD}$ soporta una mayor variabilidad de sus valores sin afecta al resultado de la simulación de la red genética.

En cuanto a aquellos parámetros que muestran una gran robustez en la red, resulta de interés el hecho de determinar si ésta se debe o no a compensaciones en los valores de los demás parámetros, es decir, estudiar si valores inusualmente extremos de otro parámetro relacionado provoca una disminución en la importancia del parámetro en cuestión dentro de una solución determinada. Tomemos como ejemplo a los parámetros $P_{PHOQACTK}$, que determina la fuerza de PhoQ-ACT para fosforilar a PhoP, y K_{PHOPP_mgta}, que determina la fuerza con la cual la especie PhoP-P se asocia al DNA de mgta activando su trascripción. Un valor inusualmente grande de $P_{PHOQACTK}$, que incrementaría en gran medida la concentración de PhoP-P, podría enmascarar la importancia de K_{PHOPP_mgta} haciendo que siempre se activara la trascripción de mgta, con una consecuente robustez de la solución con respecto a este último parámetro. En la figura 4.19. puede verse un análisis realizado a tal efecto, en el que se corrió la red utilizando 6 puntos equidistantes en el rango en el que el parámetro $P_{PHOQACTK}$ se mostraba robusto, es decir en el que el *score* se mantenía entre valores cercanos al obtenido en la solución original, en relación a la robustez del parámetro K_{PHOPP_mgta}. Allí se puede ver que la gráfica es similar para cada valor posible y lo único que cambia es el rango en el que este último parámetro se muestra robusto, lo cual se explica, ya que el aumento del parámetro $P_{PHOQACTK}$ aumenta la fosforilación de PHOP-P, lo cual favorece a la traducción de mgta, ya que aumenta la cantidad de PHOP-P, lo cual hace más flexible el valor que puede tomar K_{PHOPP_mgta}. Como se puede ver en la figura, dentro de los valores biológicamente significativos no hay una incidencia tan importante entre estos parámetros. Pruebas similares realizadas sobre otras especies muestran que no existen correlaciones significativas de a pares entre lo valores de los parámetros en las distintas soluciones, por lo que de existir esta compensación se debería a combinaciones de más de dos parámetros.

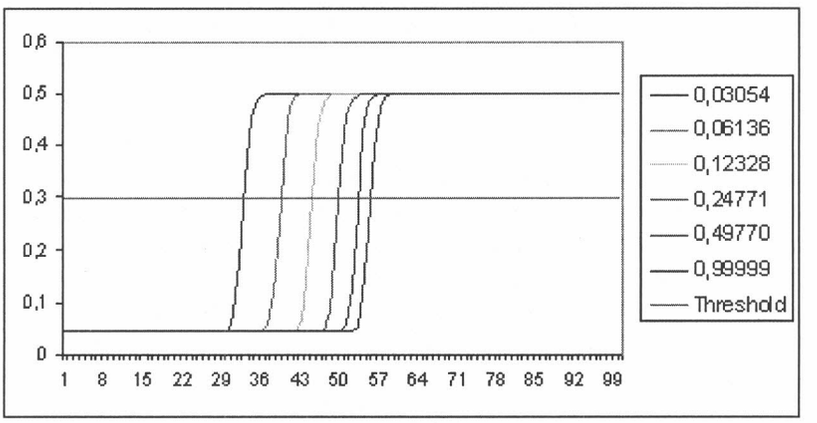


Figura 4.19. Robustez del parámetro K_{PHOPP_mgta} en relación a cambios en el valor del parámetro $P_{PHOQACTK}$ para el rango en el que se muestra robusto.

En las figuras 4.20 y 4.21 se muestran los resultados de realizar el análisis de sensitividad de los parámetros anteriormente descriptos en base a diez soluciones tomadas al azar. Los parámetros se encuentran agrupados por tipo según la descripción mostrada en el cuadro 4.1: vidas medias (prefijo H_), valor medio de activación máxima de la trascripción (prefijo K_), fosforilación (prefijo P_), dimerización (prefijo r_), coeficiente de Hill (prefijo nu_), coeficiente de saturación para el enhancer (prefijo alpha_) y otros. Para la figura 4.19 se tomó el promedio de los porcentajes del rango biológico de cada parámetro en lo que la ejecución de la simulación mantenía un puntaje por debajo del valor umbral, es decir en el que la ejecución de la red era satisfactoria. La otra figura (4.20) divide este porcentaje según distintos criterios (<5%, <25%, <50%, >75% e igual al 100%) mostrando la proporción de cada tipo de parámetro que la cumple.

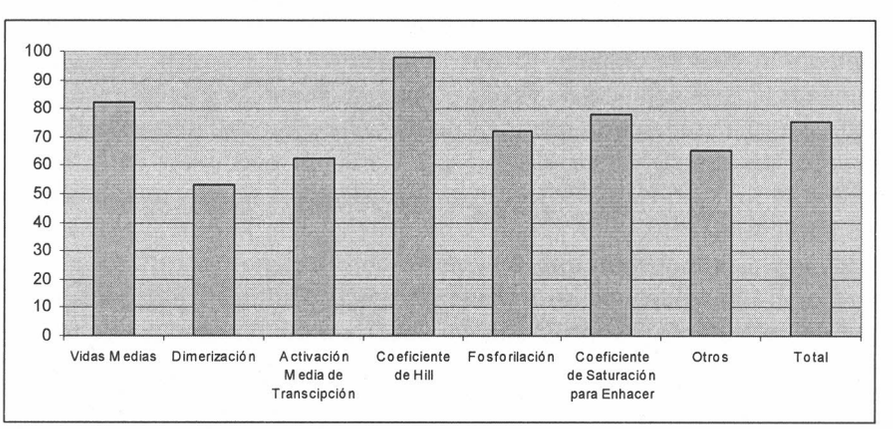


Figura 4.20. Promedio del porcentaje del rango biológicamente significativo en el que se muestra robusto cada tipo de parámetro para un conjunto de 10 soluciones tomadas al azar.

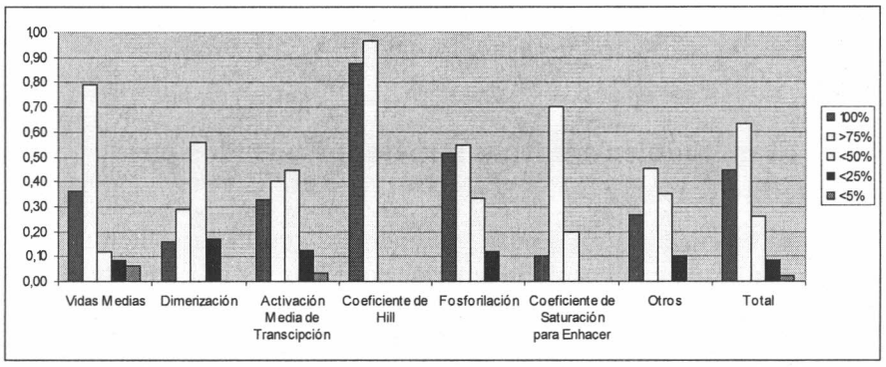


Figura 4.21. Proporciones para cada tipo de parámetro según distintos valores del porcentaje del rango biológicamente significativo en el que se muestran robustos para un conjunto de 10 soluciones tomadas al azar.

En el Cuadro 4.5 se listan aquellos parámetros puntuales que mostraron una proporción menor al 50% del intervalo biológicamente significativo en promedio de las soluciones tomadas como muestra. Si consideramos que un parámetro es poco robusto cuando el porcentaje del rango biológicamente significativo es inferior al 25%, entonces vemos que sólo hay 3 parámetros poco robustos para el promedio de las soluciones. Tengamos en cuenta, además, que la red genética utilizada consta de 66 parámetros, con lo que solo el 4,5% de los parámetros están mostrando poca robustez.

Parámetro	Prom.	Prop.	Min	Max
H_pbgp	3,3	1	3	6
K_PMRA-P_PMRD_pbgp	24,7	0,7	12	95
r_PMRA-P_PMRD	25	0,5	14	42
K_PHOQ-ACT_PHOP	29	0,3	13	38
K_PHOP-P_pmrd	29,7	0,3	3	48
max_PMRD	32,8	0,6	19	57
r_PMRA-P	34,9	0,5	5	100
K_PMRB_PMRA	37,4	0,3	2	60
r_PHOP-P	43,3	0,1	17	100
r_PHOQ	43,4	0,1	21	69
K_PHOQ_PHOP	43,4	0	32	100
r_PHOQ2PHOQ-ACT	43,6	0	33	54
K_PMRA-P_pmrd	45,5	0,1	5	74
P_PHOQ-ACT_KINASE	46,4	0,1	13	71
P_PMRB_KINASE	47	0,4	15	100
K_PHOP-P_mgta	48,9	0	43	63

Cuadro 4.5. Ranking de los parámetros que mostraron menor promedio en el porcentaje del rango biológicamente significativo (se tomaron solo aquellos con un promedio menor al 50%).

Prom: Promedio del porcentaje del rango biológicamente significativo
Prop: Proporción de los parámetros que mostraron un porcentaje menor al 25% del rango
Min, Max: Mínimo y máximo porcentaje del rango que mostró el parámetro.

3.5. El método en relación a métodos basados en búsqueda aleatoria

Según las ejecuciones realizadas y viendo los resultados obtenidos por método explicado en el capítulo anterior podemos ver el siguiente cuadro en el que se relacionan ambos resultados:

	Búsqueda Aleatoria	Algoritmo Genético
Mínimo puntaje obtenido en las simulaciones	> 0,25	0,00892
Proporción de soluciones obtenidas en relación a simulaciones ejecutadas	0,00035474	0,1796

Cuadro 4.6: Comparación entre el método basado en búsqueda aleatoria y nuestro método. Para el cálculo de la proporción de resultados se tomaron los promedios de ejecución de varias corridas.

Como se puede visualizar en el cuadro nuestro método obtiene soluciones con un menor score, lo que se traduce a una mejor configuración de la red para la simulación, lo que lleva a resultados más correlacionados con la realidad. Por otro lado, la proporción de resultados obtenidos es más grande en varios órdenes de magnitud.

En la próxima sección se verá la significación biológica de los resultados obtenidos, pero podemos adelantar que nuestro método obtiene soluciones más correlacionadas con la realidad experimental que el método basado en búsqueda aleatoria (cuadro 4.7).

Correlación de Pearson	Búsqueda Aleatoria	Algoritmo Genético
mgta	0,474	0,983383
pmrd	0,12	0,997998
pbgp	0,44	0,991515

Cuadro 4.7: Comparación entre el método basado en búsqueda aleatoria y nuestro método.

3.6. Significación biológica de las soluciones obtenidas

Una vez optimizada la búsqueda de soluciones al problema propuesto y determinada la robustez de las mismas, el siguiente paso es el de determinar la significación biológica de estos resultados. En esta sección nos concentraremos en este aspecto.

Con el fin de determinar con mayor precisión si los resultados obtenidos en las simulaciones se ajustan a lo observado experimentalmente, se realizaron ensayos de Green Fluorescent Protein (GFP)[27] que muestran el nivel actividad de PhoP como regulador de la transcripción de los genes phop, mgta y pmrd (Cuadro 4.8). Estos experimentos permiten

determinar la expresión de cada gen regulado por PhoP. Luego se le aplicaron a los valores obtenidos una función polinomial para suavizar las curvas presentadas por dichas expresiones, con el objetivo de reducir el ruido de las experimentaciones biológicas, pero manteniendo la cinética de expresividad de las mismas.

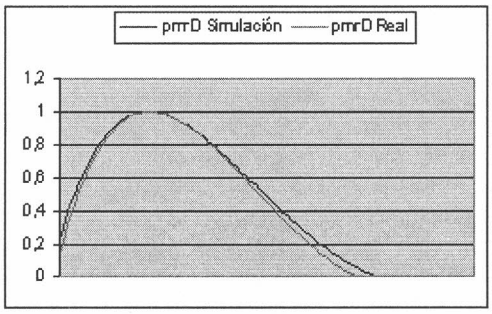
Tiempo (min)	phop	mgta	pmrd
0	3,078	1,723	0,495
5	11,632	3,218	1,236
10	21,423	4,065	2,121
20	31,214	4,502	3,527
30	28,210	4,088	3,836
60	16,878	2,358	1,547
90	19,019	2,065	1,668

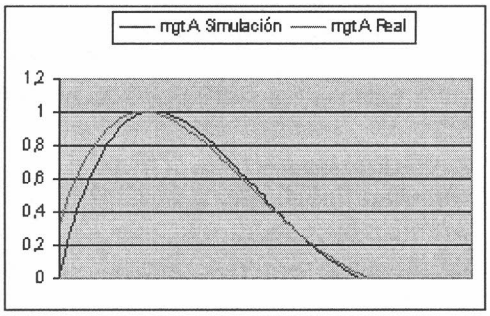
Cuadro 4.8: Valores obtenidos experimentalmente de la expresión de PhoP, mgtA y pmrD por técnicas GFP.

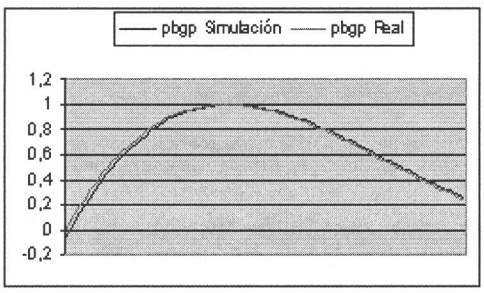
Luego se determinó la correlación entre estos resultados y los obtenidos en distintas corridas de nuestro método. Para ello se utilizaron coeficientes de correlación de Pearson (Figura 4.22). Este coeficiente refleja el grado de relación lineal entre 2 variables. Su rango va del +1 al -1, significando para el mayor valor una relación linear perfectamente positiva entre ambas variables. La fórmula para calcular dicho coeficiente es la siguiente:

$$r = \frac{\left[\frac{1}{N-1}\right]\sum(x-\bar{x})(y-\bar{y})}{S_x S_y} \qquad (4.6)$$

Los dos términos que aparecen en el denominador corresponden a los desvíos estándar de las variables cuya correlación se está midiendo (x e y). El numerador, por otro lado, calcula lo que se conoce como covarianza, lo cual representa la variación en conjunto entre ambas variables.







pmrd	mgta	pbgp
0,997998	0,983383	0,991515

Figura 4.22. Comparación de los resultados obtenidos en nuestro método en relación con los valores experimentales. El cuadro muestra la correlación de Pearson de estos valores.

En esta figura se observa la gran correlación que existen entre una de las soluciones obtenidas tomadas al azar y los resultados obtenidos experimentalmente. Para hacer este cálculo se realizó una aproximación por un polinomio de grado 5 y se normalizaron las soluciones y los resultados experimentales. Finalmente se realizó el cálculo de correlación de Pearson. También en la figura se observan las curvas de expresión de cada uno de los genes en estudio (mgta, pmrD y pbgp), observándose la similitud en el comportamiento de los mismos entre las soluciones obtenidas con nuestro método y los resultados experimentales.

4. Discusión

En el presente capítulo hemos explicado un método para el aprendizaje de redes regulatorias bacterianas basado en algoritmos genéticos. Este método tiene la ventaja de ir buscando las mejores soluciones en cada

iteración en base a mejoras en las puntuaciones que recibe cada corrida de simulación en relación al conjunto de parámetros de entrada que toma.

Además de la optimización en la obtención de las condiciones iniciales para las corridas de simulaciones, hemos visto que las soluciones obtenidas son robustas y presentan un gran realismo en relación a los valores obtenidos en experimentaciones biológicas.

El método también ha demostrado tener un mejor comportamiento que otros métodos basados en técnicas de búsqueda aleatoria, el cual fue explicado en el capítulo 3.

Capítulo 5

Conclusiones

1. Introducción

Los experimentos realizados sobre la red genética de los sistemas de dos componentes PmrA/PmrB y PhoP/PhoQ permiten extraer diversas conclusiones con respecto al método presentado en este trabajo y a éstos sistemas en particular. Desde el punto de vista de la biología celular se muestra un modelo soportado experimentalmente (biológica y computacionalmente) para la interacción a nivel transcripcional y post-transcripcional para estos dos componentes. Esta interacción permite a la célula combinar distintas funcionalidades asociadas a una misma función biológica en base a la interacción de distintos estímulos. De esta manera, se constituye una agregación que conforma una verdadera red genética, que por su simpleza relativa resulta óptima para el análisis.

Desde la óptica computacional, resulta de interés el proceso de formalización de un problema complejo derivado de la biología molecular, permitiendo transformar una serie de hipótesis biológicas (con distinto nivel de soporte experimental) en un sistema "testeable" computacionalmente, en el sentido de formular hipótesis sobre la arquitectura de las redes genéticas, así como la optimización y medición de robustez de sus parámetros para determinar las características de su funcionamiento. Esta formalización permite la formulación y verificación de hipótesis a nuevos niveles, como las referidas a propiedades inherentes a la arquitectura de los sistemas estudiados.

En las siguientes secciones plantearemos algunas conclusiones dejando para el final del capítulo algunas reflexiones acerca de futuros trabajos que pueden realizarse a partir del presentado aquí.

2. La utilización de algoritmos genéticos permite obtener mejores soluciones en las simulaciones

Como quedó en evidencia en el capítulo anterior (sección 3.5), utilizando heurísticas de optimización, como lo son los algoritmos genéticos, en la etapa de obtención de los conjuntos de valores para los parámetros de la red genética previa a la corrida de las simulaciones, se lograron mejores resultados que con técnicas de búsqueda aleatoria. Como se puede ver en los cuadros 4.6 y 4.7, esta mejora se da a tres niveles:

1. Mayor proporción de soluciones obtenidas.

2. Mejor resultado numérico para las soluciones encontradas.
3. Mayor correlación biológica para dichas soluciones

La importancia de obtener una mayor proporción de soluciones radica en el hecho de que la relación entre posibles configuraciones iniciales para las corridas y aquellas configuraciones que aseguran una ejecución correcta en la simulación, es decir, las soluciones posibles es muy grande, este hecho reduce la capacidad de las técnicas basadas en búsqueda aleatoria para hallarlas, ya que, obviamente las probabilidades de encontrar una solución es muy baja. Eso hace que se necesiten muchas corridas hasta alcanzar una configuración inicial que satisfaga a la red.

Por otro lado, como también se explicó en el capítulo anterior, cuanto menor es el score obtenido en una simulación mejor es la significación biológica de dicha solución. Nuestro método obtuvo soluciones cercanas al cero.

Finalmente, se comprobó, que la correlación entre los resultados obtenidos en las simulaciones y las experimentaciones realizadas por nuestro método fueron mucho más altas que aquellas que se lograron con el método basado en búsqueda aleatoria.

3. Existe una alta correlación entre las arquitecturas obtenidas por nuestro método y las experimentaciones biológicas

Como se puede ver en la figura 4.22 del capítulo anterior, existe una alta correlación entre los resultados obtenidos por nuestro método y las experimentaciones realizadas a nivel biológico. Este hecho es fundamental para corroborar que nuestro método realiza predicciones sobre la actividad de distintos genes regulados por la red estudiada a partir de configuraciones iniciales de parámetros de una manera que se relaciona con hechos biológicos.

Un aspecto interesante de esta característica, es que nuestro método parte de una arquitectura conocida para hacer sus predicciones sin utilizar información extra para guiar al proceso de obtención de valores para los parámetros que componen el modelo.

Por otro lado una vez obtenida (aprendida) una solución válida nuestro método va buscando las mejores variantes de parámetros a partir de los distintos operadores genéticos que utiliza el algoritmo implementado, para las siguientes generaciones. En el caso estudiado de los genes regulados por PhoP, los parámetros inciden directamente sobre el tiempo de activación, activación máxima y tiempo de inicio de la disminución de la expresión. Estas modificaciones predicen la cinética de genes cuya expresión varía en función de estas características, clasificándolos en forma automática.

4. PmrA/PmrB y PhoP/PhoQ constituyen una red genética robusta y flexible

La habilidad de la arquitectura para reproducir diversos patrones es una propiedad sistémica, relativamente independiente del valor de los parámetros escogidos. Como se observa en la figura 4.21 sólo una pequeña proporción de los parámetros del modelo tiene un límite de un rango del 25 % de valores dentro del intervalo propuesto como biológicamente significativo. Adicionalmente, una gran cantidad de parámetros puede tomar valores dentro de todo el rango propuesto (una variabilidad que se encuentra mucho más allá de lo esperable biológicamente).

Es por ello que, la habilidad de la red para reproducir patrones no se ve afectada por los parámetros individuales seleccionados, y estos pueden tomar valores dentro de un entorno de dimensiones significativas, con variaciones como las vistas en la figura 4.18 (nótese la gráfica de parámetros como H_PHOQ). Por otro lado, desde el punto de vista computacional, si bien esta robustez agrega una gran cantidad de complejidad al proceso de búsqueda de conjuntos de parámetros óptimos para la red, hemos visto que este problema puede ser resuelto utilizando algoritmos genéticos.

Finalmente, desde el punto de vista biológico, este comportamiento robusto de los parámetros puede evidenciar un fenotipo que confiere una ventaja adaptativa a lo largo de la evolución. Efectivamente, esta robustez podría permitir por un lado el mantenimiento del fenotipo global deseado a pesar de pequeños cambios (mutaciones) en el genotipo que determina la bioquímica de las especies involucradas. Por otro lado, la acumulación de pequeñas mutaciones puntuales permitiría, al transcurrir la evolución, que la red "navegara" aleatoriamente el espacio de valores posible sin afectar el fenotipo inicial, hasta alcanzar regiones en las cuales podrían convivir distintos patrones de funcionamiento (la flexibilidad observada en la red). Esto indica que el poder de la arquitectura para reproducir los patrones pedidos reside más en su topología (es decir, las especies y relaciones propuestas) que en los valores de los parámetros escogidos.

5. La agregación de los módulos PmrA/PmrB y PhoP/PhoQ genera una red genética

Las interacciones entre los módulos PmrA/PmrB y PhoP/PhoQ determinan una red genética constituida por dos subsistemas. Los modelos implementados para la interacción entre los módulos PmrA/PmrB y PhoP/PhoQ permitieron especificar muchos puntos no considerados explícitamente hasta el momento. Esto, sumado a un análisis funcional que aborda exhaustivamente el comportamiento de la

red, permite aportar evidencia desde una nueva óptica para sustentar (y eventualmente redirigir) la investigación "in vivo".

Resulta claro que modelos tales como el mostrado en las figuras 4.13 y 4.14 constituyen una hipótesis para el funcionamiento de la red, y debe considerarse que cada una de las conexiones entre los nodos (ejes numerados en la red) constituye una sub-hipótesis acerca de la interacción de los mismos (es decir, entre las especies). Asimismo, cada una de estas conexiones se implementa en el modelo mediante una cierta ecuación diferencial, escogiendo entre distintas posibilidades como se describe en [16] y [12]. El modelo computacional permite comprobar el funcionamiento de la red considerando la arquitectura como una agregación de sub-hipótesis "completa", en el sentido de que se consideran todos los elementos necesarios para su funcionamiento. Esto contrasta con la experimentación "in vivo" en la cual, tanto por costos como por desconocimiento de los mecanismos puntuales involucrados, se evalúa al sistema como una agregación de ciertos componentes escogidos sin formular hipótesis formales sobre el funcionamiento de las partes no evaluadas.

Las hipótesis de arquitectura para la red no tendrían valor si no estuvieran acompañadas por hipótesis de funcionamiento (como ya se ha expuesto, valores de concentración iniciales y finales para ciertas especies). Si bien el funcionamiento observado experimentalmente para la red es conocido, al formular todos los patrones esperables para la propuesta interacción de los subsistemas se consigue una visión a nivel de red del problema, permitiendo poner a prueba todas las componentes del sistema en un mismo experimento.

Por lo tanto, la suma de los modelos presentados, la evidencia experimental analizada y los resultados obtenidos (determinando el realismo, flexibilidad y robustez del sistema) muestra que la funcionalidad de la red puede explicarse en base a la interacción de dos módulos interconectados tanto a nivel transcripcional como post-transcripcional.

6. Trabajos futuros

El método presentado en este trabajo ha permitido comprobar las asunciones que lo motivaron. Sin embargo hay ciertos aspectos que fueron surgiendo que motivarían futuros trabajos. En esta sección se enumeran algunos de ellos.

Como se explicó en el capítulo anterior (sección 2.3.2), la naturaleza del problema a resolver es multiobjetivo y la solución propuesta utiliza una aproximación de no dominancia, como es el cálculo por suma pesada. Este método presenta ciertas desventajas ya que los distintos objetivos se combinan para formar un solo. Sin embargo se podrían utilizar otros métodos de optimización multiobjetivo, como por ejemplo, los que tiene en cuenta frentes de pareto diverso, como lo hace el algoritmo NSGA-II[5].

Por otro lado, nuestro método utiliza como función de aptitud un valor obtenido al finalizar la corrida de la simulación, a partir de las concentraciones finales de cada especie de nuestra red. Otra mejora que se podría incorporar a nuestro método es la posibilidad de ir midiendo la función de aptitud en relación al comportamiento de las distintas especies a lo largo de la simulación y no al finalizar ésta. Dicha mejora nos permitiría controlar oscilaciones en la expresión de las distintas especies que le restarían significación biológica a algunas soluciones que se pueden obtener a partir del modelo estudiado.

Finalmente se podrían estudiar otros métodos de optimización para comparar resultados.

Bibliografía

[1] Back T, Fogel D, and Michalewicz Z, editores. Handbook of Evolutionary Computation. IOP Publishing Ltd., Bristol, UK, 1997.

[2] Batchelor E, and Goulian M. Robustness and the cycle of phosphorylation and dephosphorylation in a two-component regulatory system. PNAS, 100(2):691–696, Jan 2003.

[3] Brenner S. The end of the beginning. Science, (287):2173–2174, Mar 2000.

[4] Darwin C. On The Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. John Murray, London, 1859.

[5] Deb K, Pratap A, Agarwal S, and Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6:182-197, 2002.

[6] Dongwoo S, Zwir I, and Groisman E. Positive autoregulation of the two-component system phop/phoq triggers sequential activation of the phop regulon. 2003.

[7] Fuhrman S, and Liang S. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. Pacific Symposium on Biocomputing, pages 18–29, 1998.

[8] Groisman E, Heffron J, Hoch J.A, and Silhavy T.J. Two component signal transduction. Am. Soc. Microbiol. Press., 1995.

[9] Harari O, Romero-Zaliz R, Rubio-Escudero C, and Zwir I. Fusion of domain knowledge for dynamic learning in transcriptional networks. IDEAL 2006. Proceedings to be published by Springer Verlag in the prestigious Lecture Notes in Computer Science (IDEAL=Intelligent Data Engineering and Automated Learning).

[10] Harari O, Rubio-Escudero C, and Zwir I. Targeting genes by multi-objective and multimodal shoots. 2006.

[11] Haupt R.L, and Haupt S.E. Practical Genetic Algorithms. Ed. Wiley, 1998.

[12] Kaern M. Regulatory dynamics in engineered gene networks. 4th International Systems Biology Conference, Washington University, St. Louis. 2003.

[13] Deb K. Multi-objective Optimization using Evolutionary Algorithms. Ed. Wiley, 2001.

[14] Kato A, Latifi T, and Groisman E. Closing the loop: The pmra/pmrb two-component system negatively controls expression of its posttranscriptional activator pmrd. PNAS, 100:4706–4711, 2003.

[15] Lee T, and Rinaldi N. Transcriptional regulatory networks in saccharomyces cerevisiae. Science, 298(5594):799–804, Oct 2002.

[16] Meir E, Munro E, Odell G, and Von Dassow G. Ingeneue: a versatile tool for reconstructing genetic networks, with examples form the segment polarity network. Journal of experimental zoology, 294(3):216–51, Oct 2002.

[17] Meir E, Von Dassow G, Munro E, and Odell G. Robustness, flexibility, and the role of lateral inhibition in the neurogenic network. Current Biology, 12(10):778–786, May 2002.

[18] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskil D, and Alon U. Network motifs: simple building blocks of complex networks. Science, 298(5594):824–827, Octubre 2002.

[19] Mjolsness E, Mann T, Castaño R, and Wold B. From coexpression to coregulation: an approach to inferring transcriptional regulation among gene classes from large-scale expression data. Advances in neural information processing systems, 12:928–934, 2000.

[20] Passarge E. Color Atlas of Genetics, 2nd Edition. Ed. Thieme. 2001.

[21] Ronen M, Rosenberg R, Shraiman B, and Alon U. Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. Proceedings of the National Academy of Sciences of the United States of America, 99(16):10555-60, 2002.

[22] Traverso P. Aprendizaje de dinámicas robustas en redes regulatorias bacterianas utilizando GENIE. Tesis de Licenciatura en Ciencias de la Computación. Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. 2004

[23] Von Dassow G, Meir E, Munro E, and Odell G. The segment polarity network is a robust developmental module. Nature, 406(6792):188–192, Jul 2000.

[24] Wahde M, Hertz J, and Andersson M.L. Reverse engineering of sparsely connected genetic regulatory networks. 2nd Workshop on Computation of Biochemical Pathways and Genetic Networks, Heidelberg, June 2001.

[25] Zwir I, Huang H, and Groisman E. Analysis of differentially-regulated genes within a regulatory network by GPS genome navigation. Bioinformatics 21(22):4073-83, 2005.

[26] Zwir I, Shin D, Kato A, Nishino K, Latifi T, Solomon F, Hare JM, Huang H, and Groisman E. Dissecting the PhoP regulatory network of Escherichia coli and Salmonella enterica. PNAS. 102(8):2862-2867, 2005.

[27] Zwir I, Shin D, Romero Zaliz R, Huang H, and Groisman E. Identifying the promoter features governing differential expression of co-regulated genes with similar network motifs. 2006.

[28] Zwir I, Traverso P, and Groisman E. Semantic-oriented analysis of regulation: the phop regulon as a model network. International Conference on Systems Biology, 2003.