Tesis defendida por Victor Manuel Cervantes Salido

y aprobada por el siguiente Comité

Dr. Israel Marck Martínez Pérez
Director del Comité

Dr. Carlos Alberto Brizuela Rodríguez
Miembro del Comité

Dr. Jorge Olmos Soto
Miembro del Comité

Dr. José Antonio García Macías
Coordinador del Programa de
Posgrado en Ciencias de la Computación

Dr. David Hilario Covarrubias Rosales
Director de Estudios de Posgrado

14 de enero del 2013

# CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA


---

**Programa de Posgrado en Ciencias**

**en Ciencias de la Computación**

---

Optimización de sondas en autómatas moleculares para el diagnóstico y tratamiento de fibrosis quística

Tesis

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

Maestro en Ciencias

Presenta: Victor Manuel Cervantes Salido

Ensenada, Baja California, México

2013

Resumen de la tesis de Victor Manuel Cervantes Salido, presentada como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación. Ensenada, Baja California.

Optimización de sondas en autómatas moleculares para el diagnóstico y tratamiento de fibrosis quística

Resumen aprobado por:

Dr. Israel Marck Martínez Pérez

Director de Tesis

El cómputo biomolecular es un área interdisciplinaria que tiene como finalidad la construcción de dispositivos moleculares en base a concentraciones de moléculas orgánicas que ante un estímulo externo se auto-ensamblen y se organicen de manera programada y lógica como respuesta a ese estímulo. Una de las aplicaciones potenciales de esta área es en medicina molecular, principalmente en el diagnóstico y tratamiento de enfermedades a nivel molecular. En este trabajo se diseñan mecanismos de sensado basados en ADN que sean capaces de identificar la presencia de mutaciones genéticas en moléculas de ARN. Dichos mecanismos se utilizan como complejo de diagnóstico en un modelo que implementa autómatas moleculares conocidos como gen computacional. Como aplicación hipotética, se eligió la enfermedad de la fibrosis quística producida por la mutación $\Delta F508$, y de esta manera, se propone un mecanismo teórico para el diagnóstico y tratamiento de dicha enfermedad. El problema se plantea como un problema de optimización multi-objetivo, en donde se desea minimizar una serie de funciones de energía que entran en conflicto con el tamaño de los mecanismos prepuestos, satisfaciendo al mismo tiempo un conjunto de restricciones ambientales tales como el nivel de pH y la temperatura de incubación. La optimización se realiza utilizando meta-heurísticas, específicamente mediante el uso de algoritmos genéticos. No obstante, debido a las características del problema, fue necesario diseñar un cromosoma especial que permitiera la representación de estructuras secundarias en moléculas de ADN y ARN. De igual manera, se proponen nuevos operadores de cruzamiento y de mutación compatibles con esta representación. El desempeño de los complejos se verifica por medio de un ambiente simulado *in silico* utilizando métodos Monte Carlo, específicamente el algoritmo de Gillespie. Pensando en una futura aplicación de los diseños, se generaron dos modelos de simulación, un modelo termodinámico de diagnóstico encargado de medir la especificidad y sensibilidad de los complejos, y un modelo de diagnóstico y tratamiento con el cual se simula el comportamiento del gen computacional.

Palabras Clave: **Cómputo biomolecular, optimización de sondas, simulación estocástica, autómatas moleculares.**

Abstract of the thesis presented by Victor Manuel Cervantes Salido, in partial fulfillment of the requirements of the degree of Master in Sciences in Computer Science . Ensenada, Baja California.

Probe optimization in molecular automaton for diagnostic and treatment of cystic fibrosis

Biomolecular computing is an interdisciplinary field whose purpose is to build molecular devices based on concentrations of organic molecules, which at an external stimulus they self-assemble and organize in a planned and logical way in response to that stimulus. One potential application of this area is in molecular medicine, particularly in diagnosis and treatment of diseases at molecular level. In this thesis a DNA-based sensing mechanism capable of identifying the presence of mutated genetic transcripts is designed. The proposed mechanism will be used as a diagnostic complex for a molecular automaton known as computational gene. As a hypothetical application, the cystic fibrosis $\Delta F508$ gene mutation is studied, and thus a theoretical mechanism for diagnosis and treatment of such disease is generated. The problem is formulated as a multi-objective optimization task, in which different objective functions such as the Gibbs free energy and the length of DNA complexes are in conflict. The optimization is performed by using meta-heuristics, specifically genetic algorithms. However, due to the nature of the problem, a particular chromosome that allows representations of DNA and RNA secondary structures is presented. Similarly, operators of mutation and crossover compatible with this representation are proposed. The performance of complexes were tested using simulated environments (*in silico*) through Monte Carlo methods, specifically the Gillespie algorithm. Thinking of a future implementation of the design, two simulation models, a thermodynamic model to measure the diagnostic specificity and sensitivity of the complex, and a model which simulates the behavior of the computational gene are generated.

Keywords: **Biomolecular computing, probe optimization, stochastic simulation, molecular automata.**

**Dedicatorias**

***A mis padres, Victor y Alba. A mis hermanas, Dulce y Diana.***

**Agradecimientos**

A mis padres, mi orgullo y ejemplo a seguir, gracias por su apoyo y cariño.

A mi director de tesis, el Dr. Israel Marck Martínez Pérez, que fue mi guía durante este camino.

A mi comité de tesis, por sus observaciones y comentarios para mejorar mi investigación.

A todos mis compañeros de la generación 2010, por su compañía y tolerancia durante mi estancia.

A los investigadores del posgrado en ciencias de la computación por su gran enseñanza académica.

Al personal del departamento de ciencias de la computación por hacer amena mi estancia en la institución.

Al Centro de Investigación Científica y de Educación Superior de Ensenada.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar esta maestría.

**Lista de símbolos**

| | |
|---|---|
| `A` | Adenina |
| `C` | Citosina |
| `G` | Guanina |
| `T` | Timina |
| `U` | Uracilo |
| ADN | Ácido desoxirribonucleico |
| AG | Algoritmo genético |
| ARN | Ácido ribonucleico |
| ARNAsaH | Ribonucleasa H |
| ATP | Adenosín trifosfato |
| Complejo Dx/TX | Molécula de ADN semi-complementaria |
| Dx | Diagnóstico |
| FQ | Fibrosis quística |
| pH | Potencial de hidrógeno |
| poly(Y) | Región rica en pirimidinas |
| Tx | Tratamiento |

**Contenido**

Página

Página

Página

## Lista de Figuras

Figura Página

**Lista de Tablas**

Tabla Página

# Capítulo 1. Introducción

## 1.1. Antecedentes y motivación

El cómputo biomolecular es un área interdisciplinaria que combina conocimientos de diferentes ciencias como la biología, la física, las matemáticas, las ciencias computacionales y la nanotecnología. La finalidad de esta área es la construcción de dispositivos moleculares en base a concentraciones de moléculas orgánicas que ante un estímulo externo se auto-ensamblen y se organicen de manera programada y lógica como respuesta a ese estímulo. El cómputo biomolecular ha demostrado capacidades de paralelismo y almacenamiento masivo aplicados en la resolución de problemas NP-completo(Adleman, 1996). Sin embargo, uno de los nichos en el que se vislumbra mayor aplicación es en medicina molecular, principalmente en el diseño de computadoras moleculares encargadas de detectar y tratar enfermedades a nivel celular. Aunque en la literatura se han propuesto varios modelos teóricos los cuales demuestran alta eficiencia en cuanto a exactitud y especificidad en la detección de enfermedades provocadas por mutaciones genéticas aberrantes, estos se pueden mejorar al utilizar sondas optimizadas en sus mecanismos de detección (Martínez-Pérez, 2007; Benenson *et al.*, 2001, 2004; Graugnard *et al.*, 2010).

## 1.2. Definición del problema

En el problema del diseño de sondas para el diagnóstico de mutaciones aberrantes mediante genes computacionales, se busca una configuración de complejos de ADN para diagnóstico/tratamiento, tal que se maximice el número de verdaderos positivos, esto es, maximizar la detección correcta de moléculas de ADN o ARN mutadas, mientras se

minimizan el número de falsos positivos o el número de detecciones incorrectas. Además, es necesario que el complejo Diagnóstico/Tratamiento sea termodinámicamente estable a 37°C, la temperatura promedio del cuerpo humano.

## 1.3. Objetivos

### 1.3.1. Objetivo general

El objetivo de la tesis es diseñar mecanismos de sensado basados en ADN capaces de identificar la presencia de mutaciones genéticas en moléculas de ARN. Este mecanismo se utilizará como complejo de diagnóstico en el modelo de genes computacionales. Además, como caso de prueba se eligió la enfermedad de la fibrosis quística producida por la mutación $\Delta F508$ (Vega-Briceño, 2004), y de esta manera, generar un mecanismo teórico para el diagnóstico y tratamiento de dicha enfermedad.

### 1.3.2. Objetivos específicos

Analizar la teoría relacionada con la optimización de sondas genéticas.

Analizar la fisiopatología de la fibrosis quística, y la genética relacionada con la misma.

Proponer un modelo para la representación y optimización de sondas.

Desarrollar un algoritmo para la generación de sondas mediante estrategias evolutivas. Dicho algoritmo se utilizará para diseñar el complejo de diagnóstico.

Simulación del proceso de sensado del complejo de diagnóstico, con la finalidad de corroborar el correcto desplazamiento de las hebras.

Simulación del ensamblamiento del gen computacional.

## 1.4. Metodología

Para el cumplimiento de los objetivos, primero se plantea el problema como un problema de optimización multi-objetivo, por lo que se especifican las diferentes funciones y restricciones del problema.

Posteriormente, la optimización se realiza utilizando meta-heurísticas, principalmente mediante el uso de algoritmos genéticos. No obstante, debido a las características del problema, fue necesario diseñar una representación compatible con las estructuras físicas de los moléculas ADN y ARN. De esta manera, basándose en la representación utilizada en el software DSD (Phillips y Luca, 2009), se diseña el individuo a utilizar en el AG como un vector lineal de tamaño variable, donde la información contenida se encuentra dada por un lenguaje diseñado de tal manera que pueda representar estructuras de complejos ADN/ADN y ADN/ARN (e.g. regiones complementarias, bucles interiores, bultos superiores e inferiores, nucleótidos sobresalidos). De igual manera, se diseñaron operadores de mutación y cruzamiento compatibles con la representación, los cuales respeten las restricciones de los individuos (e.g. la posición de la sección no complementaria con la que se realizará la detección), por lo que se presentan 2 nuevos operadores de cruzamiento y 3 de mutación basados en la literatura.

Con la finalidad de verificar el desempeño de los complejos, sin la necesidad de realizar físicamente experimentos en laboratorio, se propone realizarlos por medio de un ambiente simulado *in sillico* utilizando métodos Monte Carlo, específicamente el algoritmo de Gillespie (Gillespie, 1977). Pensando en una futura aplicación de los complejos optimizados, se generaron dos modelos de simulación, un modelo termodinámico de diagnóstico encargado de medir la especificidad y sensibilidad de los complejos, y un modelo de diagnóstico y tratamiento con el cual se simula el comportamiento del gen

computacional.

## 1.5. Organización de la tesis

El documento se organiza de la siguiente manera: el Capítulo 2 introduce teoría en materia de biología molecular, la cual constituye la esencia del cómputo biomolecular. De manera similar, en el Capítulo 3 se definen los fundamentos teóricos en computación que se utilizarán en los capítulos subsecuentes. Este provee los conceptos básicos en teoría de autómatas, optimización mono y multi-objetivo, dominancia y optimalidad de Pareto, conceptos de algoritmos evolutivos, algoritmos para la resolución de problemas de optimización multi-objetivo, el algoritmo NSGA-II y teoría de simulación estocástica. El Capítulo 4 introduce cómputo biomolecular, antecedentes, teoría, diferentes modelos y aplicaciones modernas en el área de medicina.

El resto del trabajo se ocupa de desarrollar las contribuciones al estado del arte. En el Capítulo 5 se describe el problema del diseño de sondas para el diagnóstico de mutaciones aberrantes, principalmente en la detección de la mutación $\Delta F508$. Además, se desarrolla el como se atacó el problema de optimización multi-objetivo, definiéndose la representación utilizada en el individuo, parámetros del algoritmo genético, operadores de cruce y mutación, implementación del algoritmo en el paquete OPT4j; así como la metodología utilizada para la búsqueda y selección de mejores individuos. El capítulo 6 desarrolla los modelos de simulación estocástica utilizados para medir el desempeño de los complejos Dx/Tx optimizados, se describen los resultados de simulación de cada modelo; así como también su correspondiente análisis. Por último, el capítulo 7 concluye el trabajo de tesis con una pequeña discusión sobre los resultados obtenidos, así como trabajo a futuro.

# Capítulo 2. Fundamentos de biología molecular

## 2.1. La célula

La célula es la unidad fundamental estructural y funcional de los organismos. Un organismo puede estar constituido por una sola célula, llamados organismos unicelulares, o puede estar conformado por varias, clasificados como organismos pluricelulares. Estructuralmente la célula consta de un compartimento de interior acuoso, llamado citoplasma, rodeado por una capa impermeable formada principalmente de lípidos denominada membrana. Dentro del citoplasma se encuentran diferentes compartimentos u organelos, los cuales están encargados de las funciones de supervivencia de la misma y estos difieren entre organismos. Las células se clasifican en dos tipos: procariotas y eucariotas. Las células eucariotas contienen un compartimiento interno donde se almacena el material genético, llamado núcleo. En contraste las células procariotas carecen de este compartimiento. Pero sin importar el tipo de célula, ambas almacenan genética en el ácido desoxiribonucleico (ADN), el cual es un polinucleótido que se replica y transmite durante la división celular. El ADN no puede ser expresado a proteínas sin antes ser convertido en moléculas intermedias (ARN), las cuales conforman la plantilla de síntesis de estas proteínas. Tanto ADN, ARN y proteínas están conformadas por bloques fundamentales, entre los cuales encontramos azucares, nucleótidos y aminoácidos. Sin embargo, el trabajo de síntesis de estas moléculas se imposibilita sin la existencia de energía, la cual es aportada por la célula en forma de adenosín trifosfato (ATP), fuente de energía de todos los seres biológicos. El ATP se libera en la célula, por ejemplo en eucariotas por medio de la mitocondria, en un proceso de transformación química de moléculas ricas en energía, tales como azucares.

## 2.2. Los ácidos nucleicos

Los ácidos nucleicos son polímeros cuyos monómeros o unidades fundamentales son los nucleótidos. Un nucleótido es una molécula conformada por una base nitrogenada, ya sea purina o pirimidina, diferenciadas solamente por la presencia de un azúcar (ya sea ribosa o desoxirribosa) y un grupo fosfato. Dentro de los ácidos nucleicos se encuentra el ADN y ARN, moléculas que se diferencían entre sí por la estructura física, las bases nitrogenadas que lo conforman y el tipo de azúcar utilizado en la columna vertebral. En los ácidos nucleicos, los nucleótidos se encuentran entrelazados uno a otro por medio de enlaces covalentes entre el grupo fosfato de un nucleótido y el grupo hidroxilo del nucleótido vecino, lo que conforma la columna vertebral, conocido como enlace fosfodiester. Es por eso que los ácidos nucleicos se encuentran en cadenas polarizadas que inician en el grupo 5'-fosfato y terminan con un grupo 3'-hidroxilo.

### 2.2.1. ADN

El ADN o ácido desoxirribonucleico es el encargado de transmitir la información hereditaria necesaria para la construcción de células u organismos. Se caracteriza químicamente por utilizar desoxirribosa como molécula de azúcar en la columna vertebral y por las bases nitrogenadas que lo conforman. Estas bases son adenina (`A`), guanina (G), citosina (`C`) y timina (`T`), las cuales se agrupan según su estructura física en purinas (adenina y guanina) y pirimidinas (timina y citosina). Según el modelo de Watson y Crick, el ADN está conformado por cadenas doblemente entrelazadas de manera antiparalela por medio de enlaces no covalentes generados por la interacción entre una base nitrogenada purina con una pirimidina. Dichos enlaces se conocen como puentes de hidrógeno, y pueden variar según el par de bases nitrogenadas: mientras que el número

de puentes generados entre las bases guanina y citosina es de tres, las bases adenina y timinina generan dos. Aunque la estructura más estable del ADN es en doble cadena (dsADN), también es posible encontrarlo en cadenas sencillas (ssADN), principalmente durante la fase de replicación de ADN en la célula y en algunos virus. Durante la fase de replicación, un ssADN sirve como plantilla para generar una hebra complementaria y crear una molecula ADN de doble hélice. Este proceso se realiza mediante la propiedad de complementariedad de las bases, en la que una base timina es complementaria a una base adenina, mientras una base guanina es complementaria a una citosina.

### 2.2.2. ARN

Al igual que el ADN, el ARN o ácido ribonucleico está conformado por cuatro nucleótidos unidos covalentemente, pero se diferencía por el azúcar utilizado en su columna vertebral, ribosa en lugar de desoxiribosa, y además por la base nitrogenada uracilo (`U`) que se intercambia por la base nitrogenada timina (`T`). De igual manera, se respetan las mismas reglas de complementariedad, por lo que la base uracilo (`U`) se complementa con la base adenina (`A`) formando dos puentes de hidrógeno. Estructuralmente, el ARN se representa por medio de una cadena sencilla, que gracias a sus propiedades termodinámicas, puede adoptar estructuras secundarias y terciarias capaces de catalizar reacciones químicas.

## 2.3. Los genes

La función más importante del ADN es almacenar la información que determina las proteínas y moléculas de ARN que constituyen un organismo, incluyendo información de cuando, en que células y la cantidad de proteínas que deben generarse. Toda esta información está organizada dentro de la célula en paquetes de secuencias llamados

genes. Un gen se define como un segmento de ADN, el cual contiene las instrucciones necesarias para sintetizar moléculas de ARN que después serán traducidas en proteínas. Está compuesto por una región no codificadora, encargada de ser una señal de paro con la cual delimita al gen, una región reguladora que controla la producción de proteínas y una región codificadora que especifica la estructura primaria de la proteína. A toda la información genética codificada en ADN de un organismo se le conoce como genoma. En células procariotas y algunos organelos de células eucariotas los genes están organizados en dsADN empaquetados de manera circular. Cada gen inicia con un sitio promotor (la región reguladora), seguida por la región codificadora de la proteína. Esta información está bien delimitada por dos regiones cortas de ADN llamadas codones de inicio y fin. El codón de inicio está especificado por la secuencia `ATG`, en cambio el codón de fin por una de las tripletas `ATA`, `ATC` o `ATT`. De manera contraria, los genes en las células eucariotas se empaquetan por medio de proteínas de forma lineal en cromosomas, los cuales se localizan en el núcleo de la célula. Una de las propiedades importantes de los genes en las células eucariotas es que un mismo gen puede codificar diferente proteínas. Esto es mediante la combinación de secuencias codificadoras (exones) y eliminando las secuencias no codificadoras (intrones).

## 2.4. El dogma central de la biología

El dogma central de la biología define el proceso de transcripción-traducción de ADN a proteínas. Originalmente se pensaba que este proceso se realizaba en un solo sentido, sin embargo con el descubrimiento de los retrovirus se modificó totalmente el esquema. La transcripción es el proceso por el cual las moléculas de ADN se transcriben a ARN, por medio de una batería de enzimas y proteínas, y esta se realiza en regiones

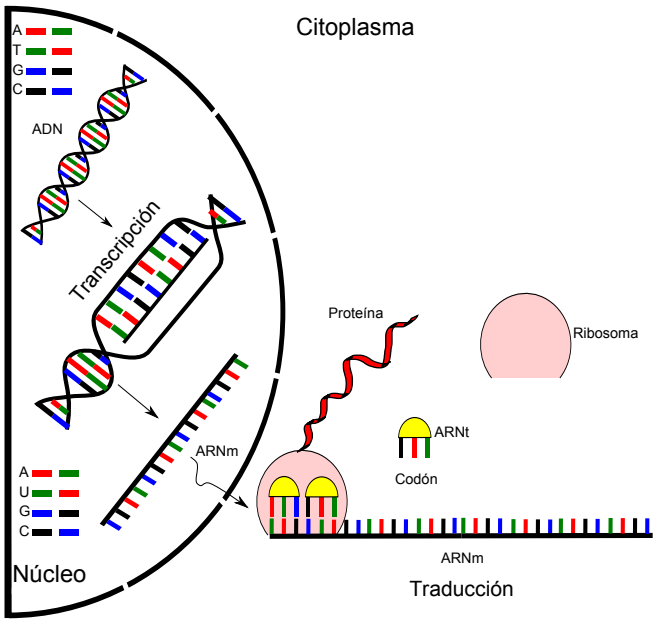



Figura 1: El dogma central de la biología.

codificadoras de genes delimitados por sus sitios promotores y de terminación. Es en el sitio promotor donde la enzima ARN polimerasa (ARNpol) inicia con la transcripción, sintetizando una molécula de ARN, llamada ARN mensajero (ARNm). Aunque todas las células de un individuo contienen la misma información genética, cada una de ellas expresa solamente ciertas regiones necesarias para realizar la función inherente de la misma. Además, pese a que el ADN se encuentra tanto en celulas eucariotas y procariotas, la expresión de proteínas se realiza de diferentes maneras. La traducción es el proceso por el cual la información codificada en ARN es sintetizada en aminoácidos, los cuales conforman proteínas. Esta traducción se lleva a cabo de acuerdo al código genético, el cual involucra palabras consistente en tripletes de nucleótidos o codones, con la característica principal de ser degenerado, es decir, uno o mas codones pueden representar un mismo aminoácido (redundante), pero ninguno representa otro aminoácido (no ambiguo). La maquinaria celular encargada de este proceso involucra diferentes tipos de ARN y una gran variedad de proteínas.

## 2.5. Mutaciones

Pese a que el sistema de replicación, transcripción y traducción de genes es un proceso con baja tendencia a errores, éstos no se encuentran exentos de producir alguno. A estos tipos de errores se les conocen como mutaciones. En bacterias, la tasa de mutación corresponde a una por cada 300 generaciones, esto es, una en $\sim 10^5$ por gen por generación ó una en $\sim 10^9$ por par de bases por generación. Actualmente no existe un cálculo de estas tasas en eucariotas, pero se piensa que es similar a la de las bacterias. Las causas de una mutación pueden ser diversas, entre las que destacan:

Extrínsecas o inducidas

Radiación: e.g. radiación UV, rayos X, radioactividad.

Agentes mutagénicos: e.g. gas mostaza, ácido nitroso, bromouracilo, acrifina, LSG, nicotina, cafeína.

Intrínsecas o espontáneas

Errores durante la replicación o recombinación de genes.

Alteración química de bases nitrogenadas, debidas a "parasitismo" genético, e.g. inserción de retrovirus transposones.

Cualquiera de estos factores puede producir mutaciones génicas, las cuales modifican la información contenida en el ADN, produciendo una expresión errónea de la información codificada. Las mutaciones génicas se pueden clasificar como:

Sustituciones. Aquellas generadas por reemplazos de nucleótidos. Dentro de este grupo encontramos las transversiones y transiciones que a su vez, pueden ser sinónimas (no modifican aminoácidos) o no sinónimas (modifican aminoácidos).

Borrados e inserciones. Estas mutaciones pueden borrar o agregar una o varias bases, produciendo un corrimiento de cuadro (siempre y cuando el borrado o la inserción no sea múltiplo de tres), el cual borra uno o varios aminoácidos produciendo una proteína aberrante.

Inversiones. Son aquellas en donde una sección entera de ADN se invierte. Una inversión es pequeña si involucra pocas bases en un gen; mientras una inversión grande involucra varias regiones de un cromosoma, el cual contiene varios genes.

Translocaciones. Este tipo de mutaciones ocurre cuando una secuencia se mueve de lugar. Esto sucede cuando la célula, en un intento de reparar bases dañadas, une terminales de ADN no contiguas.

## 2.6. Proteínas

Una proteína es un polímero conformado de unidades fundamentales llamados aminoácidos, los cuales se encuentran unidos de manera lineal por medio de enlaces covalentes conocidos como enlaces peptídicos. En total existen 20 aminoácidos esenciales, que combinados entre sí, pueden formar millones de proteínas diferentes. Una proteína puede adoptar una estructura tridimensional determinada por su secuencia de aminoácidos. Además, puede unirse a otras moléculas por medio de regiones particulares llamadas sitios de pegado. Esta propiedad permite a la proteína formar complejas estructuras supramoleculares capaces de catalizar reacciones moleculares, generar estructuras celulares, permitir movimiento, sensar señales, etc.

## 2.7. Enzimas

Las enzimas son proteínas que sirven como catalizadores de reacciones químicas dentro de la célula. Varios modelos de cómputo por ADN han utilizado enzimas dentro de operaciones básicas. Entre las más utilizadas están las enzimas de restricción, que fungen como mecanismos de defensa en la célula, principalmente en organismos procariotes. Estas enzimas pueden reconocer ciertas regiones de ADN, y una vez encontradas, las cortan, generando subconjuntos de secuencias. Otra enzima de gran utilidad es la ligasa, que ayuda a la formación de enlaces fosfodiester entre cadenas sencillas de ADN. Otras enzimas especialmente útiles son las exonucleasas y endonucleasas, las cuales cortan nucleótidos encontrados al final de la cadena de ADN o en una localidad interna, respectivamente.

## 2.8. Termodinámica y cinética molecular

Una característica que permite la complementariedad Watson-Crick es, que si una cadena ssADN se expone con otra cadena ssADN complementaria, se formará una cadena dsADN. A este fenómeno se le conoce como hibridación de ADN. Aunque una cadena ssADN no sea totalmente complementaria con otra, las bases pueden interactuar según su complemento Watson-Crick para formar cadenas dobles imperfectas. Al grado de interacción entre los nucleótidos se le conoce como rigor de hibridación. En general, mientras la temperatura de reacción de hibridación aumente, el rigor aumentará. A la temperatura en la cual más de la mitad de la concentración de moléculas dsADN totalmente complementarias se separan en ssADN se conoce como temperatura de fusión, $Tm$ ($^{\circ}C$). En condiciones de bajo rigor, las hebras pueden hibridar con más pares de bases que en condiciones de alto rigor. Una unidad para medir este rigor es la energía

libre de Gibbs, $\triangle G^\circ$, la cual otorga la condición de equilibrio y de espontaneidad de una reacción química. Esta energía se libera como calor en un evento de hibridación, el cual se puede calcular de la siguiente manera:

$$\triangle G^\circ = \triangle H^\circ - T\triangle S^\circ, \tag{1}$$

donde $\triangle H^\circ$ el cambio en la entalpía (la facilidad con la que un sistema absorbe o libera calor) y $\triangle S^\circ$ es la entropía (medida de desorden del sistema). Actualmente existen modelos para la predicción de hibridación utilizando parámetros termodinámicos según el tipo de complejo, ya sea ADN/ADN (SantaLucia y Hicks, 2004), ADN/ARN (Sugimoto *et al.*, 1995; Watkins *et al.*, 2011) o ARN/ARN (Xia *et al.*, 1998). Sin embargo, el modelo más utilizado es el del vecino más cercano (SantaLucia y Hicks, 2004), debido a su facilidad de cálculo y su exactitud.

### 2.8.1. Energía libre de Gibbs

La energía libre de Gibbs en una molécula dsADN dada por $x = a_1 \dots a_n$, y su cadena correspondiente reversa complementaria $\bar{x} = \bar{a_n} \dots \bar{a}_1$ se calcula como:

$$\Delta G^\circ = \Delta g_i + \Delta g_s + \sum_{i=1}^{n-1} \Delta G^\circ(a_i a_{i+1}/\bar{a}_i \bar{a}_{i+1}), \tag{2}$$

donde $\Delta g_i$ es la energía de iniciación de hélice, $\Delta G^\circ(a_i a_{i+1}/\bar{a}_i \bar{a}_{i+1})$ es la energía generada por el dúplex $a_i a_{i+1}/\bar{a}_i \bar{a}_{i+1}$ y $\Delta g_s$ es la energía de corrección por simetría, las cuales se obtienen mediante la Tabla 1. Por ejemplo, dada la siguiente molécula de ADN

$$5'\text{–}\texttt{GCAATGGC}\text{–}3'$$

$$3'\text{–}\texttt{CGTTACCG}\text{–}5'.$$

Tabla 1: Termodinámica del vecino más cercano. Las unidades para $\Delta G^\circ$y $\Delta H^\circ$ son en $kcal/mol$ y las unidades de $\Delta S^\circ$ es en $cal/K$ por mol de interacción. La correción por simetría es aplicada solamente a dúplex auto-complementarios. La penalización por terminal aplica a cada dúplex que finalice con la secuencia `AT`. Un dúplex con ambas terminales cerradas por pares `AT` tiene una penalización de $+1.0\,kcal/mol$ para $\Delta G^\circ$.

| Interacción | $\Delta H^\circ$ | $\Delta S^\circ$ | $\Delta G^\circ$ |
|---|---|---|---|
| `AA/TT` | -7.6 | -21.3 | -1.00 |
| `AT/TA` | -7.2 | -20.4 | -0.88 |
| `TA/AT` | -7.2 | -21.3 | -0.58 |
| `CA/GT` | -8.5 | -22.7 | -1.45 |
| `GT/CA` | -8.4 | -22.4 | -1.44 |
| `CT/GA` | -7.8 | -21.0 | -1.28 |
| `GA/CT` | -8.2 | -22.2 | -1.30 |
| `CG/GC` | -10.6 | -27.2 | -2.17 |
| `GC/CG` | -9.8 | -24.4 | -2.24 |
| `GG/CC` | -8.0 | -19.9 | -1.84 |
| Iniciación | +0.2 | -5.7 | +1.96 |
| Penalización por terminal `AT` | +2.2 | +6.9 | +0.05 |
| Correción de simetría | 0.0 | -1.4 | +0.43 |

La energía libre de Gibbs está dada por:

$$\begin{aligned}\Delta G^\circ =&\Delta g_i + \Delta g_s + \Delta G^\circ(\texttt{GC/CG}) + \Delta G^\circ(\texttt{CA/GT}) + \Delta G^\circ(\texttt{AA/TT}) \\ &+ \Delta G^\circ(\texttt{AT/TA}) + \Delta G^\circ(\texttt{TG/AC}) + \Delta G^\circ(\texttt{GG/CC}) + \Delta G^\circ(\texttt{GC/CG}) \\ =&1.96 + 0.0 - 2.24 - 1.45 - 1.00 - 0.88 - 1.44 - 1.84 - 2.24 \\ =&-9.13\,kcal/mol.\end{aligned}$$

♢

### 2.8.2. Temperatura de fusión

La tempatura de fusión es la temperatura en la cual más de la mitad de las secuencias de ADN en solución son separadas. Para una solución de ADN, la temperatura de fusión

está dada por

$$T_m = \frac{\triangle H^\circ}{\triangle S^\circ + R \ln\left(\left[C_T\right]/z\right)}, \tag{3}$$

donde $\triangle H^\circ$ el cambio en la entalpía, $\triangle S^\circ$ es la entropía, $R$ es la constante de los gases, $[C_T]$ es la concentración en moles y $z$ es igual a 4 para secuencias no complementarias e igual a 1 para secuencias complementarias.

### 2.8.3. Cinética química

La cinética de cualquier proceso físico-químico está dado según sus constantes de reacción. Una reacción química es un proceso cerrado en el cual el resultado es una conversión de sustancias químicas. Las sustancias iniciales involucradas en una reacción son llamadas reactantes, las cuales interactuan entre sí llevándose a cabo un cambio químico llamado producto. Este proceso se define de la siguiente manera. Sea $X_i$ una mezcla homogénea espacial de $m$ reactantes $1 \leq i \leq m$, los cuales reaccionan para generar una mezcla $Y_j$ de $n$ productos $1 \leq j \leq n$. La reacción puede ser descrita formalmente por la siguiente ecuación (Ignatova *et al.*, 2008)

$$\alpha_1 X_1 + \ldots + \alpha_m X_m \xrightarrow{k} \beta_1 Y_1 + \ldots + \beta_n Y_n, \tag{4}$$

donde $\alpha_i$ y $\beta_j$ son los coeficientes estequiométricos con respecto a $X_i$ y $Y_j$. La velocidad de la reacción está dada por la ecuación

$$r = k\left[X_1\right]^{\alpha_1} \cdots \left[X_n\right]^{\alpha_n}, \tag{5}$$

donde $r$ es la velocidad de reacción en $M/s$, $k$ es la constante de velocidad de reacción y $[X_i]$ es la concentración en $mol/l$ del reactante $X_i$. La constante de velocidad de

reacción $k$ es principalmente afectada por la temperatura de reacción $T$ descrita por la ecuación de Arrhenius (Ignatova *et al.*, 2008)

$$k = \kappa e^{-E_o/RT}, \tag{6}$$

donde $\kappa$ es el factor de frecuencia de colisión, $E_o$ es la energía de activación en $kcal/mol$ necesaria para que la reacción química pueda suceder y $R$ es la constante de los gases. El orden de una reacción química es la suma de las concentraciones en la ecuación de velocidad de reacción, por lo que el orden de una reacción (Ecuación 4) está dado por $\alpha = \sum_i \alpha_i$. Por lo general el orden de una reacción se determina experimentalmente.

#### 2.8.3.1. Cinética química determinística

En este esquema de análisis, las ecuaciones de reacción química se tratan de una manera matemática por lo que se traducen a ecuaciones diferenciales ordinarias. Para esto, se tiene que suponer que existe un número suficiente de moléculas que puede aproximarse como una cantidad variable continua la cual varía determinísticamente sobre el tiempo. De esta manera, la ecuación química se puede describir como un sistema acoplado de ecuaciones diferenciales para la concentración de cada una de las sustancias, en términos de la concentración de todas las demás:

$$\frac{d\left[X_i\right]}{dt} = f_i\left(\left[X_1\right], \ldots, \left[X_n\right]\right), \quad 1 \leq i \leq n. \tag{7}$$

#### 2.8.3.2. Cinética química estocástica

El enfoque determinístico supone que las reacciones químicas evolucionan de manera continua y determinística sobre el tiempo, por lo que falla a la hora de capturar la

naturaleza discreta y estocástica de reacciones químicas en pequeñas concentraciones. Un ejemplo de esto son los procesos intra-celulares, los cuales se llevan a cabo en concentraciones extremadamente bajas.

Este tipo de procesos pueden ser descritos mediante una sola ecuación diferencial conocida como la **ecuación maestra**. Supóngase un contenedor de volumen $V$ que contiene una mezcla de $n$ substancias químicas distribuidas uniformemente, las cuales pueden interactuar mediante $m$ reacciones químicas específicas. Este sistema químico puede representarse con la función de probabilidad $P(X_1, \ldots, X_n : t)$, la cual define la probabilidad de que existan $X_i$ moléculas de la *i-ésima* sustancia en el volumen $V$ en un tiempo $t$, $1 \leq i \leq n$. El *k-ésimo* momento de la función de densidad probabilística $P$ con respecto a $X_i$, está dado por

$$X_i^{(k)}(t) = \sum_{X_1=0}^{\infty} \cdots \sum_{X_n=0}^{\infty} X_i^k P(X_1, \ldots, X_n; t), \quad k \geq 0. \tag{8}$$

El primer y segundo momento son de interés. Mientras la media $X_1^{(1)}(t)$ establece el número promedio de moléculas de la *i-ésima* substancia en el volumen $V$ en el tiempo $t$, la desviación cuadrática media que ocurre en este promedio está dado por

$$\Delta_i(t) = \sqrt{X_i^{(2)}(t) - \left|X_i^{(1)}(t)\right|^2}, \tag{9}$$

por lo que se espera encontrar entre $X_i^{(1)}(t) - \Delta_i(t)$ y $X_i^{(1)}(t) + \Delta_i(t)$ moléculas de la *i-ésima* sustancia en un volumen $V$ en el tiempo $t$.

La ecuación maestra describe la evolución de la función de probabilidad $P(X_1, \ldots, X_n; t)$ sobre el tiempo. Para esto, sea $a_\mu dt$ la probabilidad de que una reacción $R_\mu$ ocurra en un volumen $V$ durante el siguiente intervalo de tiempo de longitud $dt$, dado que el sistema

se encuentra en un estado $(X_1, \ldots, X_n)$ en el tiempo $t$, $1 \leq \mu \leq m$. De igual manera, sea $b_\mu dt$ la probabilidad de que el sistema experimente una reacción $R_\mu$ en un volumen $V$ durante el siguiente intervalo de tiempo $dt$, $1 \leq \mu \leq m$. Entonces la evolución con respecto al tiempo del sistema químico se puede describir por la ecuación maestra

$$P(X_1, \ldots, X_n; t + dt) = P(X_1, \ldots, X_n; t) \left[ 1 - \sum_{\mu=1}^{m} a_\mu dt \right] + \sum_{\mu=1}^{m} b_\mu dt, \tag{10}$$

donde el primer término es la probabilidad de que el sistema se encuentre en un estado $(X_1, \ldots, X_n)$ en tiempo $t$ y permanezca en ese estado durante el siguiente intervalo de tiempo $dt$. El segundo término establece la probabilidad de que el sistema sufra al menos una reacción $R_\mu$ en el siguiente intervalo de tiempo $dt$, $1 \leq \mu \leq m$. Equivalentemente la ecuación maestra puede ser escrita como

$$\frac{\delta}{\delta t} P(X_1, \ldots, X_n; t) = \sum_{\mu=1}^{m} [b_\mu - a_\mu P(X_1, \ldots, X_n; t)]. \tag{11}$$

La densidad de probabilidad $a_\mu dt$ puede ser representanda por otra densidad probabilística. Para esto, sea $h_\mu$ una variable aleatoria que especifica el número de distintas reacciones moleculares en las que puede reaccionar $R_\mu$ en el volumen $V$ en el tiempo $t$, $1 \leq \mu \leq m$. Sea $c_\mu$ la **constante de reacción estocástica** dependiente solo de las propiedades físicas de las moléculas y la temperatura del sistema, por lo que $c_\mu dt$ es la probabilidad promedio de que una combinación particular de $R_\mu$ moléculas reactantes reaccionarán en el siguiente intervalo de tiempo $dt$, $1 \leq \mu \leq m$. Entonces

$$a_\mu dt = h_\mu c_\mu dt, \quad 1 \leq \mu \leq m. \tag{12}$$

La constante de reacción estocástica depende del tipo de reacción química. Para

concluir, es importante notar que un reactante $X$ tiene $x = N_A[X]V$ moléculas en un volumen de $V$ litros, donde $N_A$ es el número de Avogadro.

#### 2.8.3.3. Conversión de concentraciones a número de moléculas

En un modelo estocástico las concentraciones se representan por un número entero de moléculas de cada especie, mientras que en un modelo determinístico, usualmente es una concentración medida en $M$ (moles por litro). Para realizar la conversión, es necesario también conocer el volumen del contenedor $V$, medido en litros. Así que, para una concentración del reactante $X$ de $[X]M$ en un volumen de $V$ litros existen $[X]V$ moles de $X$, esto es, $n_A[X]V$ moléculas, donde $n_A \simeq 6.023 \times 10^{23}$ es la constante de Avogadro (el número de moléculas en un mol).

**Reacción de orden cero.** Para la reacción

$$\emptyset \rightarrow X, \tag{13}$$

la constante de velocidad es $kMs^{-1}$, por lo que para un volumen $V$, $X$ se produce a una velocidad de $n_AkV$ moléculas por segundo. De esta manera, la constante de velocidad de reacción estocástica es solamente $c$ móleculas por segundo, por lo que se tiene

$$c = n_AVk. \tag{14}$$

**Reacción de primer orden.** Para la reacción

$$X \rightarrow ?,$$

la constante de velocidad de reacción es $k[X]Ms^{-1}$. Esto implica que para $[X]$ en un volumen $[V]$, la concentración de $[X]$ corresponde a $x = n_A[X]V$ moléculas. Dado que $X$ decrementa a una velocidad $n_A[X]V = kx$ móleculas por segundo, y la velocidad de reacción estocástica es $cx$ moléculas por segundo, se tiene

$$c = k.$$

Esto significa que para reacciones de primer orden, las constantes de velocidad de reacción estocásticas y determinísticas son siempre iguales.

**Reacción de segundo orden.** Para la reacción

$$X + Y \rightarrow ?,$$

la constante de velocidad determinística es $k[X][Y]Ms^{-1}$. Para un volumen $V$, la reacción procede a una velocidad $n_A k[X][Y]V = kxy/(n_a V)$ moléculas por segundo. Puesto que la constante de velocidad de reacción estocástica es $cxy$ moléculas por segundo, se tiene que

$$c = \frac{k}{n_A V}.$$

También es necesario considerar las reacciones de dimerización:

$$2X \rightarrow ?, \tag{15}$$

para este tipo de reacciones la constante de velocidad es $k[X]^2$, por lo que la concentración de $X$ decrementa a una velocidad $n_A 2k[X]^2 V = 2kx^2/(n_A V)$ moléculas por

segundo. Así la velocidad de reacción estocástica se expresa como:

$$c = \frac{2k}{n_A V}. \tag{16}$$

**Reacciones de ordenes superiores** Aunque es posible calcular velocidades de reacción determinísticas de ordenes superiores, estas reacciones no son comúnmente usadas es un esquema estocástico.

#### 2.8.3.4. Cinética en hibridación de ADN

Este modelo describe la reacción de hibridación entre dos ssDNA complementarias en una moléculas dsDNA, descrita por la siguiente ecuación química (Ignatova *et al.*, 2008)

$$ssDNA_1 + ssDNA_2 \underset{k_r}{\overset{k_f}{\rightleftharpoons}} dsDNA. \tag{17}$$

La reacción puede proceder en ambas direcciones, por lo que es una reacción reversible. Las constantes de velocidad $k_f$ y $k_r$ describen las reacciones de hibridación y desnaturalización, respectivamente. La constante de velocidad $k_f$ depende de la longitud de la cadena de ADN, el contexto de la secuencia y la concentración de sales:

$$k_f = \frac{k'_N \sqrt{L_s}}{N}, \tag{18}$$

donde $L$ es la longitud de la cadena más corta participando en la formación del dúplex, $N$ es el número total de pares de bases presente y $k'_N$ es la constante de velocidad de nucleación, la cual es estimada como $(4.35 \log_{10}[Na^+] + 3.5) \times 10^5$ donde $0.2 \leq [Na^+] \leq 4.0\, mol/l$. La constante de velocidad $k_r$ es muy sensible a la longitud del ADN y a la

secuencia. Se define como:

$$k_r = k_f e^{\triangle G^\circ / RT}, \tag{19}$$

donde $R$ es la constante de los gases, $\Delta G^\circ$ es la energía libre de Gibbs, generalmente obtenida mediante el modelo del vecino más cercano, y $T$ es la temperatura de incubación, la cual *in vitro* es usualmente llevada a una temperatura $T = T_m - 298.15\,K$, donde $T_m$ es la temperatura de fusión.

## 2.9. Fibrosis quística

La fibrosis quística (FQ) es la enfermedad autosómica recesiva letal más común en la población caucásica. Causada por una mutación en el gen CFTR (*Cystic fibrosis transmembrane conductance regulator*) identificado en 1989 (Rommens *et al.*, 1989; Riordan *et al.*, 1989), el cual codifica una proteína de 1480 aminoácidos encargada de regular un canal de cloro (Cl-) expresado en varias células epiteliales. La mutación más común, presente en el 66 % de los pacientes, es la producida por la deleción de un codón, la cual produce la pérdida de un residuo de fenilalanina en la posición 508 (Vega-Briceño, 2004).

En la actualidad se han encontrado alrededor de 1914 mutaciones diferentes[1], las cuales se pueden clasificar en 5 clases basadas en sus efectos en la expresión de la proteína CFTR o su función: mutaciones de tipo clase I-III, que se caracterizan por causar enfermedades severas fenotípicas, y las clases IV y V, que tienden a relacionarse con enfermedades intermedias, aunque no de forma sistemática para enfermedades pulmonares.

La sintomatología principal de la FQ es la producción de un esputo o mucosa muy

[1]Datos tomados de la base de datos de fibrosis quística: `http://www.genet.sickkids.on.ca/StatisticsPage.html`

espeso, el cual dificulta la limpieza ciliar normal, produciendo un medio de cultivo para bacterias, las cuales inducen una respuesta inflamatoria que terminan por destruir el tejido pulmonar. Los principales patógenos, como se menciona en Vega-Briceño (2004), que habitan las vías áreas de pacientes de FQ son *Staphylococcus aureus*, *Haemophilus influenza* y *Pseudoma aeruginosa*.

Debido a que la enfermedad se expresa en células epiteliales, otros órganos afectados por la FQ son el páncreas, el tracto digestivo y el aparato reproductor. Al afectar al tracto digestivo, el paciente tiende a la mala absorción de vitaminas y nutrientes produciendo malnutrición. Además, al afectar el páncreas, el paciente adquiere un tipo especial de diabetes con características parecidas a la diabetes tipo 1 y 2. En cuanto a los síntomas en el aparato reproductor, la gran mayoría de los pacientes afectados por FQ presentan infertilidad, donde las causas difieren entre hombres y mujeres.

Al ser una enfermedad genética, no existe un proceso farmacológico como tal para la FQ, por lo que se utilizan diferentes tratamientos para cada uno de los síntomas presentados. Entre los procesos utilizados se encuentran: 1) drogas que incrementan el nivel de expresión de proteínas CFTR sintetizadas, 2) correctores de CFTR que incrementan el tráfico de la proteína fuera del retículo endoplasmático y 3) potencializadores de CFTR que corrigen los defectos en las compuertas sobre la membrana celular.

## 2.10. Sensibilidad y especificidad en pruebas clínicas

Una forma de confirmar o refutar la presencia de alguna enfermedad es mediante el uso de pruebas clínicas. Idealmente una prueba correcta identifica a los pacientes con una enfermedad presente, y de igual manera, identifica a los pacientes que no son afectados por ella. En pocas palabras, una prueba perfecta nunca es positiva ante pacientes

libres de una enfermedad y nunca es negativa ante un paciente que está enfermo. Sin embargo, la mayoría de las pruebas clínicas están lejos de este objetivo. Para entender la utilidad de una prueba clínica es necesario comprender algunos conceptos como:

1. Verdadero positivo. El paciente tiene una enfermedad $x$ y la prueba es positiva.
2. Falso positivo. El paciente no tiene una enfermedad $x$, pero la prueba es positiva.
3. Verdadero negativo. El paciente no tiene la enfermedad y la prueba es negativa.
4. Falso negativo. El paciente tiene la enfermedad, pero la prueba es negativa.

Cuando se evalua una prueba clínica, los términos sensibilidad y especificidad son utilizados, y estos son independientes de la población de interés sometidos a la prueba. Por otro lado, existen valores predictivos que consideran a la población a la que se somete la prueba, como lo son el valor predictivo positivo (VPP) y valor predictivo negativo (VPN). Para términos prácticos de este trabajo, se decidió utilizar la sensibilidad y la especificidad como métricas de evaluación.

### 2.10.1. Sensibilidad

La sensibilidad de una prueba clínica se refiere a la habilidad de la prueba para identificar correctamente aquellos pacientes con alguna enfermedad. La sensibilidad se calcula como:

$$\text{Sensibilidad} = \frac{\#\text{ Verdaderos positivos}}{\#\text{ Verdaderos positivos} + \#\text{ Falsos negativos}}. \tag{20}$$

Una prueba con una sensibilidad del 100 %, identifica correctamente a todos los pacientes con la enfermedad de interés. De igual manera, una prueba con una sensibilidad

del 80 % detecta 80 % de los pacientes con dicha enfermedad (verdaderos positivos) pero 20 % de los pacientes con la enfermedad no son detectados (falsos negativos). Es por eso que una sensibilidad alta es importante donde la prueba es utilizada para detectar enfermedades serias pero tratables (e.g. cáncer de mama).

### 2.10.2. Especificidad

La especificidad de una prueba clínica se refiere a la habilidad de la prueba de identificar correctamente aquellos pacientes sin ninguna enfermedad presente. La especificidad se calcula mediente la fórmula:

$$\text{Especificidad} = \frac{\#\text{ Verdaderos negativos}}{\#\text{ Verdaderos negativos} + \#\text{ Falsos positivos}}. \tag{21}$$

Por lo tanto, una prueba con una especificidad del 100 % identifica correctamente todos los pacientes sin la enfermedad. Por otro lado, una prueba con especificidad del 80 % reporta correctamente al 80 % de los pacientes sin la enfermedad como prueba negativa (verdaderos negativos), pero reporta incorrectamente al 20 % de los pacientes sin la enfermedad como prueba positiva (falso positivo).

Una prueba con alta sensibilidad pero baja especificidad resulta en muchos pacientes sanos detectados de manera positiva, generando complicaciones a los mismos. Aunque la situación ideal (pero irrealista) es una prueba 100 % exacta, una buena alternativa es someter a los pacientes a una primera prueba con alta sensibilidad y baja especificidad, y después someterlos a una segunda prueba con baja sensibilidad y alta especificidad. De esta manera, la mayoría de los falsos positivos pueden ser correctamente identificados como enfermedad negativa (Lalkhen y McCluskey, 2008).

# Capítulo 3. Fundamentos de computación

## 3.1. Autómatas

Los autómatas de estados finitos (FSA) estudiados inicialmente en 1940 y 1950, son un tipo sencillo de máquina. En un inicio fueron propuestos como un modelo del funcionamiento del cerebro. Sin embargo hoy en día son utilizados principalmente en varios componentes de hardware y software. Un autómata puede ser visto como una unidad de procesamiento, la cual lee una cadena de entrada aceptándola o rechazándola. Los autómatas finitos se clasifican en determinísticos, no determinísticos y estocásticos.

### 3.1.1. Autómata de estados finitos determinístico (DFA)

Un autómata de estados finitos determinístico (DFA) es una quíntupla $M = (\Sigma, S, \delta, s_0, F)$ donde $\Sigma$ es el alfabeto, $S$ es un conjunto finito de estados con $S \cap \Sigma = \varnothing$, $s_0 \in S$ es el estado inicial, $F \subseteq S$ es el conjunto de estados finales, y $\delta : S \times \Sigma \to S$ es la función de transición, donde la transición $\delta(s, a) = s'$ también puede ser gráficamente representada como $s \xrightarrow{a} s'$. El tamaño de un autómata de estados finitos $M$, denotado por $|M|$, está dado por el número $|S| + |\delta|$.

Ejemplo 3.1: Sea $M$ un autómata de estados finitos con un conjunto de estados $S = \{s_1, s_2\}$, un alfabeto $\Sigma = \{a, b\}$, un estado inicial $s_0$, el conjunto de estados finales $F = \{s_0\}$ y reglas de transición $\delta$ dadas por el grafo de transiciones en la Figura 2.

Suponiendose una cadena de entrada $x = aba$, el cómputo se realiza de la siguiente manera:

1. Iniciando en $s_0$, el autómata lee el primer símbolo de la cadena $x_1$ y cambia al estado $s_1$ por medio de la regla de transición $\delta(s_0, a) = s_1$.

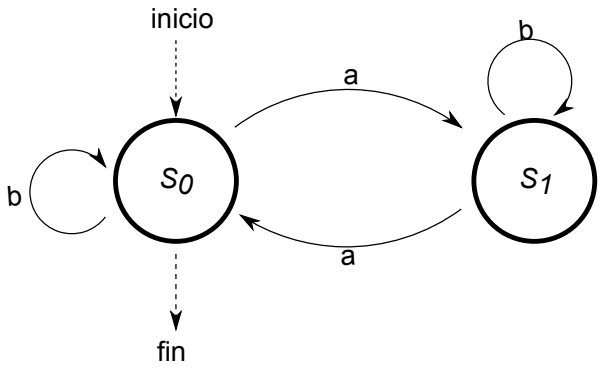



Figura 2: Autómata de estados finitos determinístico.

2. Estando en $s_1$, se lee el siguiente símbolo $x_2$ y permanece en el mismo estado $s_1$ de acuerdo a la regla $\delta(s_1, b) = s_1$.

3. Se lee el siguiente símbolo $x_3$ y el autómata cambia aplicando la regla de transición $\delta(s_0, a) = s_0$, actualizando el nuevo estado del autómata a $s_0$.

4. Al no haber más símbolos, el autómata termina el procesamiento en el estado $s_0$, aceptando la cadena.

### 3.1.2. Autómata de estados finitos estocástico

Un autómata de estados finitos estocástico (FSA) es una generalización de las máquinas determinísticas (Ignatova *et al.*, 2008). Las transiciones en una máquina estocástica están basadas en distribuciones de probabilidad. Una distribución de probabilidad $p$ en un conjunto finito $S$ es un mapeo $p : S \to \mathbb{R}_0^+$ tal que $\sum_{s \in S} p(s) = 1$. Formalmente, un FSA es una quíntupla $M = (\Sigma, S, P, q_0, q_f)$ tal que $\Sigma$ es el alfabeto, $S$ es un conjunto finito de estados con $S \cap \Sigma = \varnothing$, $q_0$ es la distribución de probabilidad inicial en el conjunto de estados $S$, $q_f$ es la distribución de probabilidad final en el conjunto de estados $S$, y $P$ es la distribución de probabilidad condicional tal que $P(\cdot \mid a, s)$ es la distribución de probabilidad en el conjunto de estados $S$ para cada par $(a, s) \in \Sigma \times S$.

## 3.2. Optimización mono-objetivo

En general, un problema de optimización mono-objetivo se define como: minimizar (o maximizar) $f(\mathbf{x})$ sujeto a $g_i(\mathbf{x}) \leq 0$, $i = \{1, \ldots, m\}$, y $h_j(\mathbf{x}) = 0$, $j = \{1, \ldots, p\}$, donde $\mathbf{x} \in \Omega \subseteq R^n$. Una solución minimiza (o maximiza) el escalar $f(\mathbf{x})$ donde $\mathbf{x}$ es un vector variable de decisión $n$-dimensional $\mathbf{x} = (x_1, \ldots, x_n)$ de algún conjunto $\Omega$. Nótese que $g_i(\mathbf{x}) \leq 0$ y $h_j(\mathbf{x}) = 0$ representan restricciones que tienen que ser cumplidas mientras se optimiza $f(\mathbf{x})$, mientras que $\Omega$ contiene todas las posibles $\mathbf{x}$ que pueden ser usadas para satisfacer la evaluación de $f(\mathbf{x})$ y sus restricciones. El problema para encontrar el mínimo global de cualquier función se conoce como optimización global, el cual puede definirse como:

Sea $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ una función con $\Omega \neq \emptyset$ y $\mathbf{x} \in \Omega$. El valor $f^* \triangleq f(\mathbf{x}^*) > -\infty$ es llamado el mínimo global si y solo si

$$(\forall \mathbf{x}) \in \Omega : \ f(\mathbf{x}^*) \leq f(\mathbf{x}), \tag{22}$$

donde $\mathbf{x}^*$ es una solución global mínima, $f$ es la función objetivo, y el conjunto $\Omega$ es la región factible de $\mathbf{x}$.

## 3.3. Optimización multi-objetivo

Un problema de optimización, en el que es necesario maximizar o minimizar $m$ variables de decisión y $l \geq 2$ objetivos, se define formalmente como:

$$\begin{aligned} \text{Minimizar o maximizar} \quad & Y = F(x) = (f_1(x), f_2(x), \ldots, f_l(x)) \\ \text{donde} \quad & x = (x_1, \ldots, x_n) \in X \quad \text{y} \\ & y = (y_1, \ldots, y_l) \in Y, \end{aligned} \tag{23}$$

donde $x$ es el vector de decisión, $X$ es el espacio de variables, $y$ es el vector objetivo, y $Y$ el espacio de objetivos. En optimización multi-objetivo, las funciones objetivo $F$ constituyen un espacio multidimensional comúnmente denominado $Z$, donde por cada solución $x$ en el espacio de decisión, existe un punto objetivo en el espacio definido por $F(x)$. El mapeo se realiza entre un vector de soluciones $n$-dimensional y un vector objetivo $l$-dimensional (Figura 3).

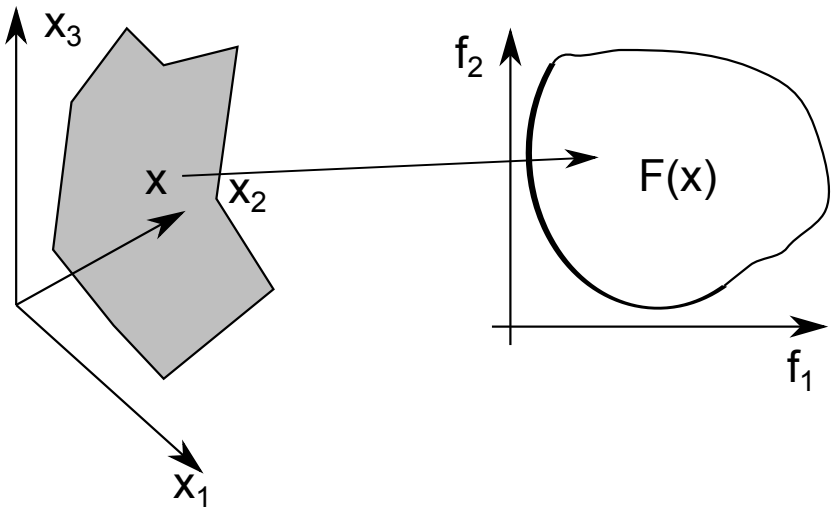



Figura 3: Mapeo del espacio de decisión al espacio objetivo.

A diferencia de la optimización mono-objetivo, la solución de un problema de optimización multiobjetivo está compuesta por un conjunto de soluciones que representan el mejor compromiso entre los objetivos. A estas soluciones se les conoce como **conjunto óptimo de Pareto**, que al ser graficados, se obtiene el **frente Pareto** del problema.

### 3.4. Dominancia y optimalidad de Pareto

Dentro de los problemas de optimización multi-objetivo existe una noción diferente de optimalidad comparado con los problemas de optimización mono-objetivo, esto es, se quiere encontrar un buen compromiso (o compensación) entre los objetivos que se

buscan optimizar. El concepto de optimalidad mayormente aceptado es la optimalidad Edgeworth-Pareto, o comúnmente conocido como optimalidad Pareto. Una solución factible $x^* \in D$ es llamada óptimo de Pareto (también llamado eficiente o no dominado) si y solo si no existe una solución $x \in D$ tal que $x$ domina a $x^*$. Una solución $y = (y_1, y_2, \ldots, y_n)$ domina a una solución $z = (z_1, z_2, \ldots, z_n)$, o de manera simbólica, $y \succcurlyeq z$, en un contexto de minimización, si y solo si $\forall i \in [1, \ldots, n],\ f_i(y) \leq f_i(z)$ y $\exists i \in [1, \ldots, l]$ tal que $f_i(y) < f_i(z)$. De esta manera, cualquier solución perteneciente al conjunto óptimo de Pareto puede ser considerada como óptima.

En un problema de minimización bi-objetivo, el frente de Pareto eficiente obtenido puede ser fácilmente graficado (Figura 3). Sin embargo, encontrar el óptimo de Pareto no siempre es una tarea trivial, haciendo necesario el utilizar meta-heurísticas para ayudar en la búsqueda de soluciones no dominadas, por ejemplo, mediante el uso de algoritmos evolutivos multi-objetivo, una meta-hueristica que se desarrollará más adelante.

### 3.5. Algoritmo evolutivo

Un algoritmo evolutivo (AE) es una heurística inspirada en la teoría de la evolución darwiniana que se utiliza ampliamente en la resolución de problemas de optimización. La idea básica de esta heurística es, dada una población de individuos en un ambiente con recursos limitados, competir por esos recursos mediante selección natural o supervivencia del más apto, resultando así en un aumento en la aptitud de la población. Dado un número de funciones de calidad a maximizar (o minimizar), se pueden generar soluciones candidatas de manera aleatoria a las cuales se les asigna una medida de aptitud mediante las funciones de calidad. De esta manera se seleccionan los mejores individuos

para construir una nueva población. Esto se efectúa aplicando recombinación y/o mutación entre ellos. La recombinación es un operador aplicado en dos o más candidatos (denominados padres) generando uno o más nuevos candidatos (hijos). El operador de mutación se aplica a un solo candidato generando un nuevo individuo. Por lo tanto, al ejecutar operaciones de cruzamiento y mutación en los padres, se genera un nuevo conjunto de candidatos (la descendencia). Durante cada ejecución, la población de padres e hijos es comparada según su aptitud, seleccionándose los mejores individuos. Este proceso continua iterativamente hasta encontrarse un candidato con suficiente calidad (una solución) o hasta encontrarse con una condición de paro, por ejemplo, el número de generaciones. Un algoritmo evolutivo está formado por dos fuerzas fundamentales:

> Los operadores de variación, recombinación y mutación, crean la diversidad necesaria dentro de la población facilitando novedad.
>
> El mecanismo de selección, el cual actúa como una fuerza que incrementa la calidad media de las soluciones en la población.

La aplicación combinada de variación y selección generalmente conlleva a la mejora de los valores de aptitud en poblaciones consecutivas. De esta manera, la aptitud no es vista como una función objetivo a ser optimizada, sino como una expresión de requerimiento ambiental. El proceso evolutivo resulta en una población la cual es gradualmente mejor adaptada a su ambiente. Cabe mencionar que la mayoría de los componentes del proceso evolutivo son estocásticos, por lo que los individuos más aptos tienen mayor probabilidad de ser seleccionados que los menos aptos, aunque estos últimos también tienen una probablidad no nula de convertirse en padres o de sobrevivir. Durante el proceso de recombinación, la elección de los puntos de cruce se realiza de manera aleatoria. De manera similar para la mutación, se escogen los alelos a mutar de un padre. El

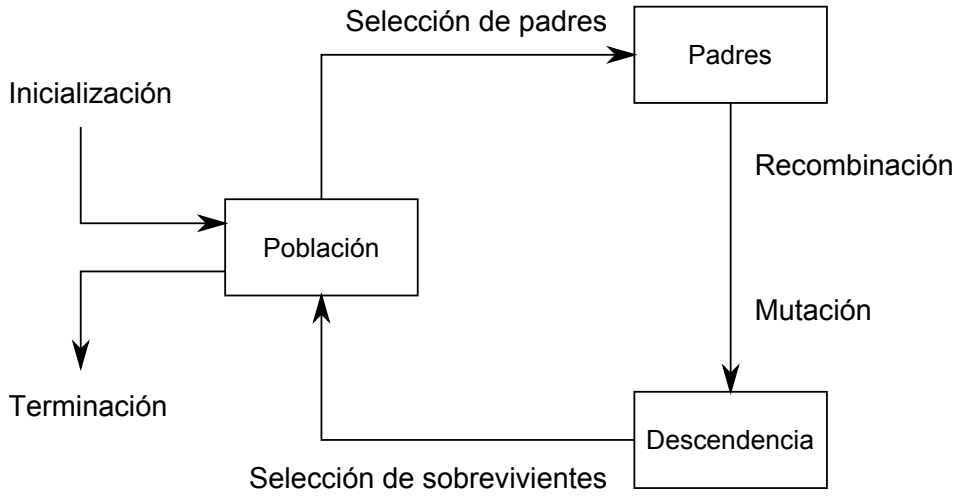



Figura 4: Esquema general de un algoritmo evolutivo.

esquema general de un algoritmo evolutivo se presenta en la Figura 4, y el pseudocódigo correspondiente se presenta en el Algoritmo 1.

Los AEs se clasifican según el dialecto o representación que utilizan los individuos. Comúnmente los indivuos en un AE son representados como cadenas sobre un alfabeto finito en algoritmos genéticos (AG), vectores de valores reales en estrategias evolutivas (EE), máquinas de estados finitos en programación evolutiva (PE), y árboles en programación genética (PG). Cabe mencionar que una representación puede ser preferente a otra si esta coincide mejor con un problema dado; esto es, permite una codificación de soluciones candidatas fácil o más natural. Por ejemplo, en el problema de satisfactibilidad con $n$ variables lógicas, la decisión directa es utilizar cadenas binarias de longitud $n$, por lo que el AE apropiado sería un altgoritmo genético. Por otro lado, si se desea evolucionar un programa de computadora capaz de jugar ajedrez, una representación por árboles es adecuada, por lo que un esquema de PG es utilizado. Como se puede observar, el utilizar diferentes representaciones requiere de operadores apropiados de recombinación y mutación que coincidan con la representación a utilizar. Por ejemplo,

en PG el operador de recombinación trabaja en árboles, mientras que en un AG opera en cadenas. A diferencia del operador de selección, el cual trabaja independiente de la representación.

**Algoritmo 1** Pseudocódigo del algoritmo genético.

**Entrada:** $M$ individuos y $N$ generaciones
**Salida:** Población de tamaño $|M|$

```
1: P←random(M)                          ▷ Población inicial de tamaño M
2: Evaluar(P)                           ▷ Evaluar la población inicial
3: para i = 0 → N hacer                 ▷ Iterar N núm. de generaciones
4:     P* ← Seleccionar(P)              ▷ Seleccionar los padres a recombinar
5:     O ← Recombinar(P*)               ▷ Recombinar pares de padres seleccionados
6:     Mutar(O)                         ▷ Mutar la descendencia
7:     Evaluar(O)                       ▷ Evaluar la nueva población
8:     P ← Seleccionar(O)               ▷ Seleccionar la siguiente generación
9: fin para
```

### 3.5.1. Componentes de los AEs

Para definir un AE es necesario especificar un número de componentes, procedimientos u operadores. Entre los más importantes se encuentran:

Representación (definición del individuo).

Función de evaluación (aptitud).

Población.

Mecanismos de selección de padres.

Operadores de variación, recombinación y mutación.

Mecanismos de selección de sobrevivientes (reemplazo).

Adicionalmente, es necesario definir un procedimiento de inicialización, el cual se encarga de generar de manera aleatoria la primera población, y una condición de terminación para generar un algoritmo ejecutable.

La **representación** es la encargada de mapear entre el contexto del problema original y el espacio de soluciones donde la evolución se lleva a cabo. Los objetos que forman las soluciones posibles dentro del contexto original del problema son conocidos como fenotipos, mientras que su correspondiente codificación en el EA se le conoce como genotipo. Al mapeo de espacio fenotípico a espacio genotípico se le llama codificación. De manera contraria, al mapeo inverso de espacio genotípico a espacio fenotípico se le conoce como decodificación. Una representación correcta tiene que ser reversible, por lo que a cada genotipo le corresponde exactamente un fenotipo.

La **función de evaluación** se encarga de representar los requerimientos a los cuales la población debe adaptarse. Formalmente, la función de evaluación es una función o procedimiento que asigna una medida de calidad a los genotipos, y típicamente se compone de una medida de calidad en el espacio de fenotipos y su representación inversa.

La **población** se encarga de guardar las soluciones posibles del problema y se representa por un multiconjunto de genotipos la cual forma la unidad de evolución. Los individuos, al contrario de la población, son objetos estáticos que no cambian ni se adaptan. Al número de soluciones diferentes dentro de una población se le conoce como diversidad.

El papel de la **selección de los padres** es el de distinguir individuos basados en su calidad, y en particular, permitir que los mejores individuos sean padres en la siguiente generación. Junto con el mecanismo de selección de sobrevivientes, la selección de padres es responsable de la mejora de la aptitud poblacional. El mecanismo de selección es

similar al operador de selección de padres, pero se utiliza en una etapa diferente del ciclo evolutivo.

Los **operadores de variación** son necesarios para generar nuevos individuos a partir de los ya existentes. El operador de **mutación** es un operador unario el cual se aplica a un solo genotipo, generando un nuevo hijo o descendencia. El operador de **recombinación** o de **cruzamiento** es un operador binario el cual combina la información de dos genotipos padres en dos o mas genotipos hijos.

Si durante la ejecución de un AE se encuentra una solución óptima conocida o dentro de un intervalo de precisión $\epsilon$, entonces se tiene una buena condición de paro. Sin embargo, en la mayoría de los problemas no se conoce con precisión cuál es la solución óptima, por lo que es necesario emplear otras condiciones. Las más comunes para esta tarea son:

Tiempo de CPU máximo permitido.

Número de evaluaciones máximas.

La mejora de la aptitud poblacional se encuentra en cierto umbral por un periodo de tiempo.

La diversidad de la población cae en un umbral dado.

## 3.6. Algoritmos para la resolución de problemas de optimización MO

La mayoría de los problemas de optimización MO pertenecen a la clase NP-difícil, por lo que los métodos exactos solo pueden ser usados en problemas de pequeña escala. Los métodos de aproximación en cambio, se utilizan generalmente para resolver problemas de grandes casos. Entre estos se encuentran algoritmos que producen una aproxi-

mación a soluciones con cierta calidad garantizada y metaheurísticas multi-objetivo las cuales pueden ser clasificadas en tres clases:

**Enfoque escalar.** En este enfoque, el problema es transformado en uno o varios problemas mono-objetivos. Los métodos de agregación (Ishibuchi y Murata, 1998), restricción-$\epsilon$ (Hertz *et al.*, 1994), métricas ponderadas, programación por metas, las funciones basadas en logro y por metas obtenidas (Coello-Coello *et al.*, 2010), son algunos ejemplos de este enfoque. Todos ellos requieren conocimiento *a priori* del problema con el fin de definir preferencias entre los objetivos, y en la mayoría de los casos, generar una sola solución por corrida.

**Enfoque basado en poblaciones.** Estos explotan la población adoptada por varias metaheurísticas (e.j., algoritmos evolutivos) con el objetivo de combinar varios procedimientos escalares en una sola corrida. Un ejemplo de este grupo es VEGA, propuesto por Schaffer (1985), el cual consiste en un algoritmo genético que utiliza tantas subpoblaciones como objetivos existentes del problema. Cada subpoblación selecciona el mejor individuo en un solo objetivo. Para esto, cada subpoblación es unida y permutada de manera aleatoria. Posteriormente se aplican operadores de cruzamiento y mutación de manera normal. La idea de este enfoque es que al recombinar buenos individuos en un objetivo, se genera un buen compromiso entre objetivos. Sin embargo, esto contradice la noción optimalidad de Pareto, por lo que es poco utilizado (Coello *et al.*, 2007).

**Enfoque basado en dominancia Pareto.** En este enfoque, el mecanismo de selección incorpora el concepto de optimalidad de Pareto. Métodos típicos dentro de esta clase adoptan un rango de soluciones basado en la optimalidad de Pareto propuesta por Goldberg (1989), con algunas variaciones: rango de dominancia

(MOGA, Fonseca y Fleming (1993)), profundidad de dominancia (NSGA-II, Deb *et al.* (2002)), y conteo de dominancia (SPEA, Zitzler y Thiele (1999) y SPEA2, Zitzler *et al.* (2003)).

**Enfoque basado en indicadores.** En este enfoque, en lugar de utilizar rango de Pareto, se utiliza una medida de evaluación de rendimiento para seleccionar soluciones. Entre ellos se encuentran IBEA (Zitzler y Künzli, 2004) y SMA-EMOA (Emmerich *et al.*, 2005).

## 3.7. NSGA-II

*Nondominated Sorting Genetic Algorithm II* (NSGA-II) (Algoritmo 2), propuesto por Deb *et al.* (2002), es un algoritmo evolutivo multiobjetivo basado en dominancia de Pareto. NSGA-II es una versión mejorada del NSGA al utilizar un operador de amontonamiento sin necesidad de parámetros en lugar de utilizar nichos. Este algoritmo utiliza un esquema de selección más (+) el cual compara la población de padres e hijos para la selección de descendencia. Además, NSGA-II permite una selección elitista y computacionalmente es más eficiente que NSGA, haciéndolo un algoritmo altamente competitivo.

## 3.8. Simulación estocástica

En ocasiones, cuando se desea entender el funcionamiento de un fenómeno natural (e.g. el proceso de transcripción y traducción de genes en células), es necesario poderlo replicar y generar datos para poderlo analizar. Es por eso que se utilizan herramientas matemáticas para modelar, y por ende, predecir un fenómeno. Sin embargo, existen casos donde la problemática es demasiado compleja o su comportamiento es aleatorio,

**Algoritmo 2** Nondomitaded Sorting Genetic Algorithm II.

**Entrada:** $M$ individuos y $N$ generaciones
**Salida:** Conjunto de soluciones no dominadas

1: $P \leftarrow \text{random}(M)$ ▷ Población inicial de tamaño $M$
2: **para** $t = 0 \rightarrow N$ **hacer** ▷ Iterar N núm. de generaciones
3: $R \leftarrow P \cup Q$ ▷ Combinar la población de padres e hijos
4: $F_R \leftarrow \text{fast\_dominated\_sort}(R)$ ▷ Obtener los frentes no dominados
5: $P_{t+1} = \emptyset$, $j = 1$
6: **mientras** $|P_{t+1}| + |F_j| \leq M$ **hacer** ▷ Iterar hasta completar $P_{t+1}$
7: $\text{crowding\_distance\_assigment}(F_j)$ ▷ Calcular la distancia de ▷ amontonamiento en $F_i$
8: $P_{t+1} \leftarrow P_{t+1} \cup F_j$ ▷ incluir el frente a la población
9: $j \leftarrow j + 1$
10: **fin mientras**
11: $\text{Sort}(F_i, \prec_n)$ ▷ Ordenar de manera descendiente ▷ Según $\prec_n$
12: $P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$ ▷ Escoger los primeros ▷ $(N - |P_{t+1}|)$ elementos de $F_i$
13: $Q_{t+1} = \text{make\_new\_population}(P_{t+1})$ ▷ Seleccionar, mutar y ▷ cruzar para generar descendencia
14: **fin para**

por lo que un análisis matemático exacto es imposible. A estos tipos de problemas se les conoce como analíticamente intratables. No obstante, esto no significa que el problema no puede ser entendido, sino que se requiere de una nueva herramienta: el análisis estocástico. Con la ayuda de una computadora, es posible simular la evolución de un fenómeno al efectuarlo varias veces y estudiando los datos percibidos. Un ejemplo de aplicación lo encontramos en el modelado de cinética en redes bioquímicas con el algoritmo de Gillespie (Wilkinson, 2006).

### 3.8.1. Algoritmo de Gillespie

En un sistema de reacciones químicas con $v$ reacciones, se sabe que la propensidad de un tipo de reacción $i$ es $h_i(x, c_i)$, donde $x$ es una especie dada y $c_i$ es la constante de

velocidad de la reacción $i$. La propensidad de que una reacción de algún tipo ocurra es:

$$h_0(x, c) \equiv \sum_{i=1}^{v} h_i(x, c_i). \tag{24}$$

El tiempo para que la siguiente reacción ocurra es $\exp(h_0(x, c))$, y es una variable aleatoria con probabilidad proporcional a $h_i(x, c_i)$ e independiente del tiempo del siguiente evento. Esto es, la reacción $i$ tendrá lugar con probabilidad $h_i(x, c_i)/h_0(x, c)$, donde es utilizado el tiempo para la siguiente reacción $j$, el sistema puede ser actualizado y la simulación puede continuar. A este tipo de simulación por eventos discretos se le conoce como el algoritmo de Gillespie. El pseudocódigo se resume de la siguiente manera:

**Algoritmo 3** Algoritmo de Gillespie.

**Entrada:** Constantes de reacción estocásticas $c_1 \ldots, c_m$, población molecular incial $x_1, \ldots, x_n$

1: $t \leftarrow 0$
2: **mientras** $t < T_{max}$ **hacer**
3: **para** $t = 0 \rightarrow N$ **hacer**
4: Calcular la propensidad $h_i(x, c_i)$ basado en el estado actual, $x$.
5: **fin para**
6: Calcular $h_0(x, c) \equiv \sum_{i=1}^{v} h_i(x, c_i)$, la propensidad de reacción combinada.
7: Simular el tiempo para la siguiente reacción, $t' \leftarrow Exp(h_0(x, c))$.
8: $t := t + t'$.
9: Simular el índice de reacción, $j$, según $h_i(x, c_i)/h_0(x, c), i = 1, 2, \ldots, v$.
10: $x := x + S^{(j)}$ ▷ donde $S^{(j)}$ denota la $j$-ésima columna ▷ de la matriz de estequiometría $S$.
11: **fin mientras**

# Capítulo 4. Cómputo biomolecular

El cómputo biomolecular es un área interdisciplinaria que combina conocimientos de diferentes ciencias como la biología, la física, las matemáticas, las ciencias computacionales y la nanotecnología. La finalidad de esta área es la construcción de dispositivos moleculares con base en concentraciones de moléculas orgánicas que ante un estímulo externo se auto-ensamblen y se organicen de manera programada y lógica como respuesta a ese estímulo. La manera en la cual estos dispositivos realizan estas dos acciones, ya sea por medio de operaciones de laboratorio o por las propiedades de auto-organización inherentes a las biomoléculas, son las diferentes vertientes de investigación del área, las cuales se explicarán más adelante.

## 4.1. El experimento de Adleman

Adleman (1994) fue el primero en explorar el poder de cómputo en biomoléculas, planteado orriginalmente por Feynman (1961), al implementar un algoritmo para solucionar uno de los problemas combinatorios difíciles, el problema del camino hamiltoniano (HPP), perteneciente a la clase NP-Completo. En el HPP, se busca encontrar un camino en un grafo dirigido iniciando y terminando en vértices específicos, visitando cada uno de los vértices en el grafo una sola vez. Al pertenecer a la clase NP-completo, actualmente no es posible solucionar este problema mediante computadoras digitales convencionales utilizando un algoritmo de fuerza bruta. Adleman, codificando el problema en cadenas de ADN y utilizando operaciones biológicas, resolvió un caso del HPP de 7 vértices en tiempo lineal. Adleman representó la información del grafo de la siguiente manera: cada vertice $v_i$ es representado por un oligonucleótido, mientras

que cada arista $e_{ij}$, la cual conecta el vértice $v_i$ al vértice $v_j$, es representado por un oligonucleótido complementario a la segunda mitad del extremo $3'$ de la secuencia de $v_i$, y de la primera mitad del extremo $5'$ del vértice $v_j$. El algoritmo de Adleman es el siguiente:

Generar una biblioteca combinatoria de soluciones.

Remover todos los caminos con vértices de inicio inválidos o con vértices finales inválidos.

Remover todos los caminos que no contienen exactamente $n$ vértices.

Remover todos los caminos con dos o más vértices repetidos.

Leer los caminos Hamiltonianos encontrados, si existen.

La biblioteca combinatoria de caminos se crea por la hibridación y unión de todos los oligonucleótidos que codifican vértices y aristas. En el paso 2, los caminos correctos se amplifican mediante PCR utilizando los primers correspondientes. Despúes, en el paso 3, los caminos que contienen exactamente $n$ vértices son extraídos por electroforesis en gel. El paso 4 requiere un ciclo de filtrado lineal en el número de vértices: por cada vértice $v_i$, el producto resultante del paso anterior es desnaturalizado, para que $v_i$ pueda hibridar, y posteriormente retenido mediante purificación por afinidad. Finalmente, el producto se amplífica utilizando PCR y las soluciones se detectan por electroforesis en gel.

## 4.2. Modelos clásicos de cómputo con ADN

Gracias a los resultados de Adleman, Lipton (1995) diseñó algoritmos basados en ADN para resolver el problema de satisfacibilidad (SAT) y otros problemas NP-

completos. Al poco tiempo, comenzaron a aparecer más aplicaciones con cómputo biomolecular utilizando el trabajo de Adleman, esquema al que se le conoce como modelo de filtrado. La idea de este modelo es generar un conjunto de bibliotecas de ADN que contenga un gran espacio de soluciones candidatas de un problema, para después filtrar todas aquellas que no sean soluciones hasta encontrar una respuesta, si es que existe. Para el problema de satisfacibilidad, Lipton definió un conjunto de operaciones básicas (i.e. extraer, unir y detectar), modelo que fue subsecuentemente refinado por varios autores (Gibbons *et al.*, 1996; Roweis *et al.*, 1998). Otro ejemplo de un problema NP-completo resuelto con modelos de filtrado es el problema del máximo clique por Ouyang *et al.* (1997). El problema se resuelve de la siguiente manera: Sea $G$ un grafo con un conjunto de vértices $V = \{v_1, \ldots, v_n\}$. Cada subconjunto de $V$ se representa por un número binario de longitud $n$, i.e. un subconjunto contiene $v_i$ si y solo si el $_i$-ésimo bit es encendido. Para esto, cada uno de los números binarios se representan utilizando cadenas sencillas de ADN de la forma $P_1V_1P_2V_2P_3V_3 \ldots P_nV_nP_{n+1}$, donde $P_i$ tiene una longitud de 20 pb y $V_i$ de 0 pb (bit 1) o 10 pb (bit 0). Estas moléculas se sintetizan utilizando la técnica POA a partir de moléculas del tipo $P_iV_iP_{i+1}$ para $i$ impar y el complemento de $P_iV_iP_{i+1}$ para $i$ par. Adémas, cada molécula $V_i$ de longitud 0 contiene un sitio de restricción entre $P_i$ y $P_{i+1}$. Estos sitios son diferentes por cada $i$ por lo que se requieren una enzima de restricción por cada vértice. El algoritmo se desarrolla de la siguiente manera:

Generar una biblioteca de subconjuntos $V$ de manera aleatoria.

Identificar cliques en la biblioteca combinatoria.

Encontrar los cliques de tamaño máximo.

Leer los máximos cliques.

El primer paso es resulta trivial, ya que solamente se necesita generar cada una de las posibles soluciones en moléculas de ADN. El segundo paso se ejecuta utilizando las enzimas de restricción. Si los vértices $v_i$ y $v_j$ son adyacentes en $G^c$, la biblioteca combinatoria se divide en dos tubos de ensayo $T_1$ y $T_2$. El contenido de estos tubos es digerido por las enzimas correspondientes a $v_i$ y $v_j$. La solución resultante es entonces mezclada en un nuevo tubo $T$. Esta operación se repite secuencialmente por cada arista en $G^c$. Las moléculas que quedan intactas despues de la reacción corresponden a cliques de $G$. Estas moléculas se amplifican mediante PCR, mientras que las moléculas incompletas no pueden ser amplificadas. En el tercer paso, las moléculas dsADN con la longitud más corta se detectan mediante electroforesis en gel, las cuales corresponden al máximo clique de $G$. Este algoritmo resuelve el problema del máximo clique en tiempo cuadrático $O(n^2)$, asumiendo que el tiempo de preparación es lineal y los pasos 1-3 son lineales en el número de vértices en $G^c$.

Aunque este modelo permite resolver problemas combinatorios en tiempo polinomial, el vólumen de ADN necesario para codificar un problema crece exponencialmente de acuerdo a su tamaño (Hartamis, 1995). Estos resultados fueron desalentadores, sin embargo el progreso de cómputo con ADN continuó y al poco tiempo, Roweis *et al.* (1998) introdujeron el modelo de etiquetas (*sticker model*) como una implementación de la máquina de registros.

## 4.3. Modelo de etiquetas

El modelo de etiquetas (*sticker model*) pertenece a los modelos de filtrado "clásicos" de cómputo basado en ADN, que se caracteriza por tener operaciones de separación o filtrado como mecanismo central de procesamiento. Este modelo es una implementación

de la máquina de registros, donde cada uno de estos consiste de una molécula sencilla de ADN de longitud fija que representa información binaria. Cada registro de información se divide en varias subcadenas (bits) de un número constante de nucleótidos. Además, existe un conjunto de etiquetas, las cuales son solamente complementarias a una subcadena del registro de datos. Si una subcadena tiene su correspondiente etiqueta hibridada, representa un bit encendido, de otra manera, representa un bit apagado. Al complejo compuesto de un registro y etiquetas se le llama complejo de memoria. De esta manera, un complejo de memoria puede representar cualquier número binario tan solo hibridando su cadena etiqueta correspondiente a la posición del bit deseado en el registro de datos. A una colección de complejos de memoria se le llama tubo, el cual puede contener múltiples copias de la misma cadena.

Aprovechando el paralelismo masivo de las moléculas de ADN, el modelo de etiquetas emplea un conjunto de operadores para manipular tubos: la combinación de dos tubos en un nuevo tubo (unión), la separación de un tubo en dos nuevos tubos (separar), y el encendido (establecer) y apagado (despeje) de un determinado bit de cada registro en el tubo. Una secuencia finita de estas operaciones se conoce como algoritmo de etiquetas (*sticker algorithm*), cuya complejidad está dada por el número total de pasos de laboratorio. Está demostrado que este conjunto de operaciones es suficientemente robusto para garantizar integridad computacional (Roweis *et al.*, 1998).

Un algoritmo de etiquetas recibe un conjunto de cadenas binarias, llamado tubo inicial, como parametro de entrada. Aunque la estrategia de la mayoría de los algoritmos propuestos es generar un tubo inicial que contenga un conjunto grande de soluciones potenciales (y despúes remover todas las no soluciones), también es posible generar un tubo inicial con soluciones aproximadas, y construir las soluciones correctas durante la ejecución del programa. Al finalizar el algoritmo, uno o más tubos finales contienen la

solución, si existe alguna.

## 4.4. Modelos autónomos de cómputo con ADN

Dentro del cómputo biomolecular, una de las líneas de investigación más activas es el diseño y construcción de autómatas finitos capaces de funcionar a escala molecular en ambientes *in vivo*. Entre los primeros modelos propuestos se encuentra el de Head *et al.* (2002), el cual implementa un autómata *in vitro* en donde el proceso de cómputo manipula un plásmido que funge como cadena de entrada. La salida del autómata está dada por el plásmido más largo al final de la manipulación. Por otro lado, Henkel *et al.* (2005) implementan un mecanismo en el que el proceso computacional se conduce en una secuencia de ADN, la cual incluye una señal de inicio de transcripción controlada a través de un promotor. El cómputo genera como resultado la construcción final de un plásmido *in vivo* conteniendo la solución computacional, la cual se construye en una proteína. Sin embargo, ambos modelos tienen la desventaja de no ser totalmente autónomos.

Es por eso que, Benenson *et al.* (2001, 2003) proponen un modelo de autómata molecular totalmente autónomo. Inspirado en el trabajo original de Rothemund (1996), el modelo de Benenson consiste en un autómata de dos estados y dos símbolos que utiliza enzimas de restricción para llevar a cabo el cómputo autónomo del mismo. El modelo está compuesto de hardware, software y entrada. El hardware es una mezcla de enzimas de restricción (endonucleasa FokI) ligasa y ATP. El software se comprende de un conjunto de cadenas doblemente enlazadas con terminales 5' sobresalientes que implementan las reglas de transición del autómata. La información de entrada también está codificada en moléculas de ADN doblemente enlazadas con terminales 5' sobre-

salientes. El modelo incorpora además dos moléculas encargadas de la detección de la salida (uno por estado). Una de las desventajas de dicho modelo es que solamente permite implementar autómatas con dos estados y un cierto número de símbolos.

Por esta razón, Kuramochi y Sakakibara (2006) proponen un modelo más general, que permite codificar $n$ estados y $m$ símbolos, conocido como autómata codificado por longitud, llamado así debido a que el número de nucleótidos necesarios para codificar el autómata está en relación con el número de estados que contiene. Sin embargo, dicho modelo tiene la limitante de que el estado inicial tiene que ser el mismo que el estado final. Martínez-Pérez *et al.* (2009) presentan su modelo de autómata de etiquetas el cual resuelve las deficiencias de los modelos anteriores al codificar reglas de transición en secuencias sencillas de ADN cortas referidas como etiquetas (*stickers*).

La primer máquina de estados finitos implementada en bacterias (*E. Coli*) la propuso Nakagawa *et al.* (2006). Este modelo está basado en un esquema de codificación dependiente de la longitud de una molécula de ARN. El procesamiento se realiza mediante la biosíntesis, combinada con técnicas artificiales que permiten la síntesis de aminoácidos a partir de codones compuestos por cuatro nucleótidos, dando como resultado una proteína de interés.

## 4.5. Modelos con autómatas moleculares aplicado al diagnóstico y tratamiento de enfermedades

Una enfermedad puede relacionarse con un conjunto de marcadores moleculares presentados (e.g. fragmentos específicos de ADN localizados en el genoma). El cáncer en particular es el resultado de la acumulación de mutaciones genéticas, las cuales provocan la pérdida de la regulación celular haciendo que estas se multipliquen descontrolada-

mente, produciendo tumores. Generalmente, esta enfermedad se detecta cuando uno o varios tumores se encuentran presentes; esto es, después de meses e inclusive años de que se originan las primeras mutaciones. De aquí la necesidad de encontrar métodos de diagnóstico a nivel molecular. En teoría, una enfermedad se puede diagnosticar por medio de un autómata finito, donde la entrada del sistema está dada por la presencia o ausencia de ciertos marcadores moleculares característicos a esa enfermedad, y la salida del autómata puede ser la generación de alguna proteína, antibiótico o señalamiento (Figura 5). Entre los trabajos relacionados con autómatas moleculares y su aplicación en medicina encontramos a Benenson *et al.* (2004) y Martínez-Pérez *et al.* (2007). En el primer trabajo se propone un autómata molecular que permite la detección y tratamiento de enfermedades, añadiendo además la capacidad de controlar la expresión del tratamiento liberado. En este modelo, el complejo de diagnóstico y tratamiento se codifican utilizando una sola secuencia de ADN o ARN. La codificación se realiza de tal manera que la secuencia de entrada produce una estructura secundaria tipo lazo. Cada cambio de estado se lleva a cabo por el uso de enzimas de restricción, las cuales cortan regiones específicas de la secuencia que define al autómata. Al finalizar el cómputo, se libera una cadena de ARN o ADN, la cual permite la expresión de una proteína correcta o actuar como droga. Este trabajo demuestra un modelo de cómputo flexible y robusto capaz de analizar lógicamente indicadores de enfermedad *in vitro* y controlar la administración biológica de tratamiento. Sin embargo, una de sus principales desventajas radica principalmente en el uso de enzimas de restricción como elementos encargados del cómputo, ya que pueden dificultar aún más su aplicación en sistemas *in vivo*.

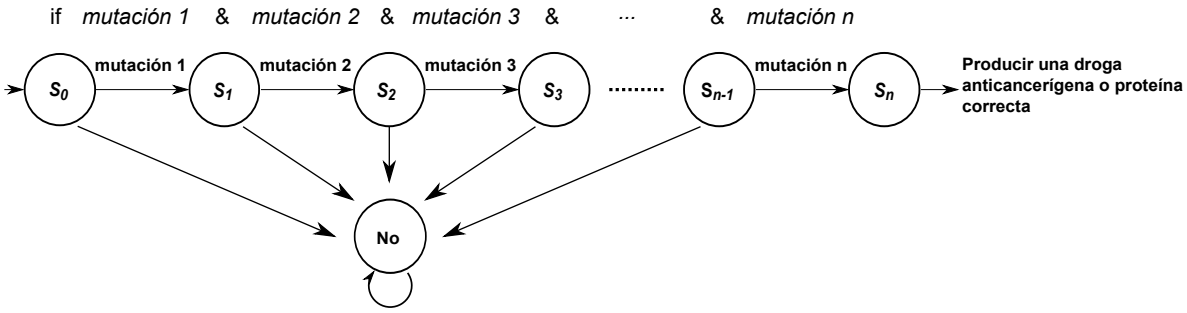



Figura 5: Autómata de estados finitos encargado de la detección de mutaciones. El autómata inicia en el estado $S_0$. Si todas las mutaciones necesarias para producir cierta enfermedad están presentes, el autómata transita de un estado $S_0$ al estado final de aceptación $S_n$, que activa la producción de un medicamento o proteína funcionalmente correcta.

## 4.6. Genes computacionales

Martínez-Pérez *et al.* (2007) presentan un modelo de autómata finito denominado genes computacionales, el cual se enfoca en la detección y tratamiento de mutaciones aberrantes en fenotipos. Un gen computacional es un gen construido artificialmente, cuyas instrucciones de ensamblamiento son generadas por un autómata de estados finitos. Está diseñado de tal manera que pueda funcionar, en teoría, en un ambiente celular, donde la presencia o ausencia de alguna condición active el auto-ensamblamiento espontáneo de un gen sintético que codifica una molécula de ARN funcional como salida. La parte estructural (Figura 6) está constituida por un gen desactivado que se utiliza como esqueleto para la codificación de la entrada y las reglas de transición del autómata. Las características conservadas del gen estructural (promotor, codones de inicio y fin de transcripción y señales de maduración de ARN) funcionan como constantes del gen computacional, mientras que las regiones codificantes, el número de exones e intrones, la posición de codones de inicio y fin, y las variables teóricas del automata (símbolos, estados y reglas de transición) son los parámetros de diseño del mismo. La desactivación se realiza cortando regiones no codificantes o intrones del gen, imposibilitando la tra-

ducción y transcripción del mismo. Para activar el gen se utilizan las hebras liberadas por el complejo diagnóstico/tratamiento, las cuales complementan las regiones no codificantes del gen, posibilitando su auto-ensamblamiento y, de esta manera, la traducción y transcripción del mismo. La parte funcional se encarga de la codificación de los estados y símbolos del autómata. La entrada del autómata son los marcadores moleculares dados en moléculas sencillas de ADN o ARN, y las reglas de transición se representan por complejos de diagnóstico/tratamiento. El complejo diagnóstico/tratamiento se encarga de la detección de mutaciones y la realización de los pasos transicionales entre estados. Si la entrada es aceptada por el autómata, se activa el auto-ensamblamiento del gen computacional, el cual será integrado en la maquinaria de traducción y transcripción celular produciendo alguna proteína o péptido que sirva como medicamento. De manera contraria, si la salida es rechazada, el auto-ensamblamiento se realiza de manera incompleta generando una mólecula que no puede ser traducida.

### 4.6.1. Diagnóstico y tratamiento de mutaciones aberrantes.

Una enfermedad relacionada con una mutación puede ser diagnosticada y tratada por la siguiente regla,

$$\begin{gathered}\text{if proteína\_X\_mutada\_en\_codón\_Y} \\ \text{then producir\_droga\_Z fi.}\end{gathered} \tag{25}$$

Esta regla podría permitir que una proteína con alguna mutación patogénica lleve a cabo su función natural. Por ejemplo, la deleción de un codón en la posición 508 del gen CFTR produce una proteína que no realiza su función correctamente, la cual es una de las causantes de la enfermedad conocida como Fibrosis quística. Sin embargo, la sintomotatología de esta enfermedad puede ser contrarrestada al recuperar la función de la

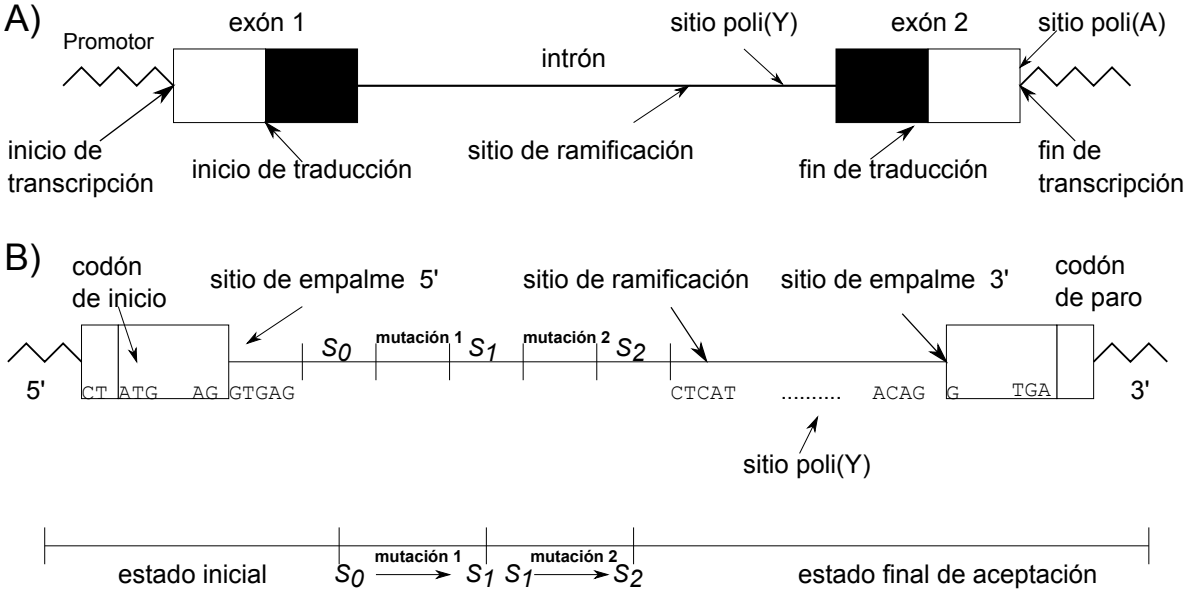



Figura 6: Diseño de un gen computacional. A) Los genes eucariotes se encuentran organizados en secciones alternas de secuencias codificantes (exones) y secuencias no codificantes (intrones). Las regiones conservadas, por ejemplo, la región rica en pirimidinas (poli(Y)) del intrón, sitios de empalme 5' (`AG/GTGAG`), dinucleótidos `AG` en el sitio de empalme 3', y secuencia de sitio de ramificación (`CTCAT`) garantizan el reconocimiento de un gen como tal por la maquinaria celular. B) Representación esquemática de un gen funcional auto-ensamblado. El estado inicial comprende al promotor, primer exón, y sitio de empalme 5', las reglas de transición se colocan en la región del intrón, y el estado final incluye el sitio de ramificación, región poli(Y), sitio de empalme 3' y el segundo exón. Adaptado de "Computational genes: a tool for molecular diagnosis and therapy of aberrant mutational phenotype", De Martínez-Pérez *et al.*, 2007, BMC Bioinformatics 8, 1 (2007), pp. 365.

proteína CFTR. La regla de diagnóstico (Ecuación 25) puede ser implementada por un autómata de dos estados y un símbolo consistente de dos móleculas dsADN parciales y una mólecula ssADN, la cual representa la presencia de la mutación relacionada con dicha enfermedad, sirviendo así como un *switch* molecular para el auto-ensamblamiento lineal del gen funcional (Figura 7). Para procesar la regla de diagnóstico, el autómata molecular debe ser capaz de detectar la mutación de interés. El encargado de este trabajo es el complejo diagnóstico/tratamiento, una molécula dsADN que se compone de una señal de tratamiento y una señal de diagnóstico (Figura 7A)). Ambas cadenas

se encuentran imperfectamente hibridadas en la región que asemeja la mutación a ser detectada. Una molécula de ARNm mutada activará la disociación del complejo e hibridará con la señal de diagnóstico. Este proceso es impulsado termodinámicamente debido a la mayor estabilidad del duplex ADN/ARN sobre el complejo ADN/ADN, debido al incremento de bases que se complementan. El complejo ADN/ARN resultante actúa como sustrato para la enzima RNasa H celular, destruyendo la secuencia de ARN del complejo. Por otro lado, la señal de tratamiento liberada se encarga de unir el gen funcional (Figura 7B)), cuya estructura es terminada por la enzima ligasa presente en células eucariotas y procariotas. La maquinaria de transcripción y traducción celular es la encargada del tratamiento y administra ya sea una proteína o un péptido como medicamento. En ambos casos, el fenotipo patogénico se suprime, ya sea al ser reemplazado por proteínas naturales o por la liberación de pequeñas moléculas que ayudan a estabilizar la proteína mutada, proporcionando la funcionalidad fisiológica de una natural.

### 4.7. Modelo de desplazamiento de hebras y su aplicación en medicina.

Dentro de las técnicas desarrolladas de cómputo molecular, el modelo de desplazamiento de hebras de ADN (DSD por sus siglas en inglés) propuesto por Yurke y Mills (2003), es un mecanismo que permite desarrollar cómputo autónomo con cadenas de ADN. En este modelo, una cadena de ADN actúa como señal, mientras que cadenas dobles o estructuras de complejos de ADN actúan como compuertas. La principal ventaja de este modelo, lo cual lo hace atractivo en la comunidad científica, es su autonomía: una vez que las señales y las compuertas son mezcladas, el cómputo procede por si solo sin necesidad de alguna intervención hasta que las compuertas o señales se agotan, y

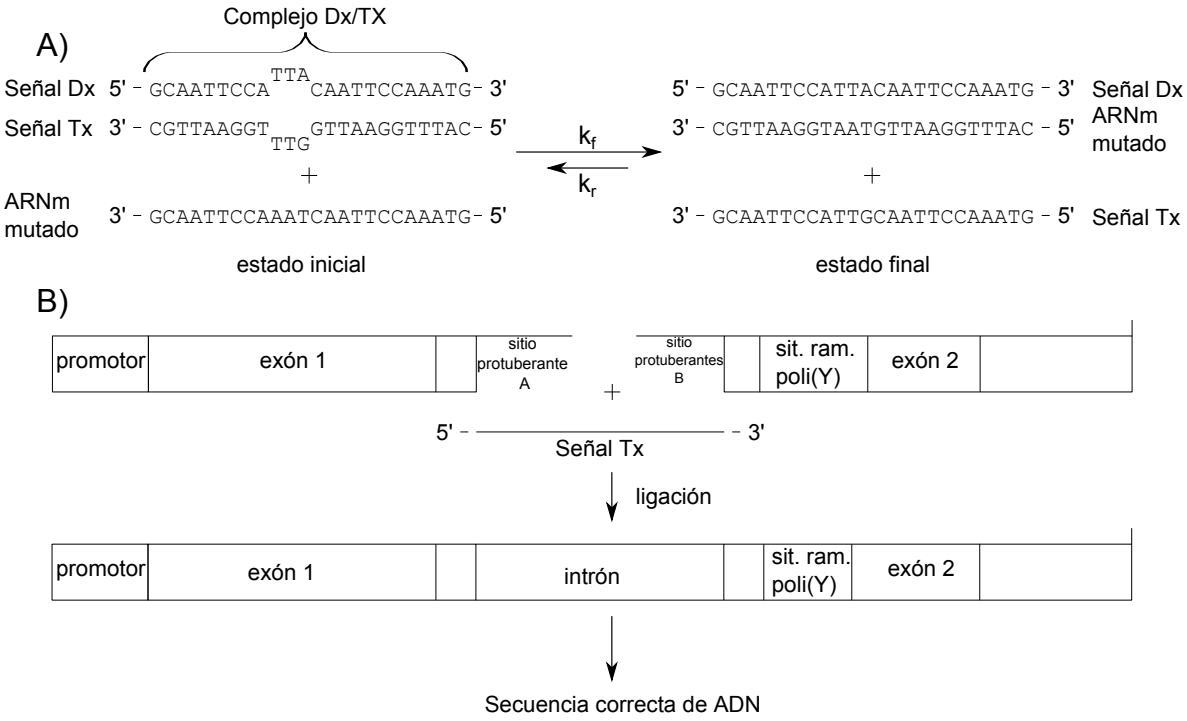



Figura 7: Modelo para el diagnóstico y tratamiento de mutaciones. A) Proceso de diagnóstico. Un complejo diagnóstico/tratamiento es una mólecula dsADN el cual asemeja una pequeña parte del gen funcional de interés, en donde una de las cadenas es intacta (señal de tratamiento Tx) y la otra contiene la mutación a ser detectada (señal de diagnóstico Dx). En el caso de una mutación patogénica, el ARNm transcrito hibrida con la señal de diagnóstico, activando la liberación de la señal de tratamiento. B) Proceso de tratamiento. La señal de tratamiento liberada complementa la estructura del gen funcional, por lo que una proteína o péptido es proporcionada por la maquinaria de transcripción y traducción celular. Adaptado de "Computational genes: a tool for molecular diagnosis and therapy of aberrant mutational phenotype", De Martínez-Pérez *et al.*, 2007, BMC Bioinformatics 8, 1 (2007), pp. 365.

cuya salida es comúnmente leída por medio de flourescencia. El mecanismo fundamental de este esquema es la migración y desplazamiento de cadenas mediada por puntos de apoyos o "*toeholds*". Este mecanismo se muestra en la Figura 8, donde cada letra y segmento correspondiente representa un dominio de ADN (secuencia de nucleótidos) y cada cadena de ADN es vista como la concatenación de varios dominios. Los dominios cortos (*toeholds*) hibridan de manera reversible con su complemento, mientras que los dominios más largos lo hacen de manera irreversible; la longitud crítica para ser

irreversible depende de las condiciones físicas.

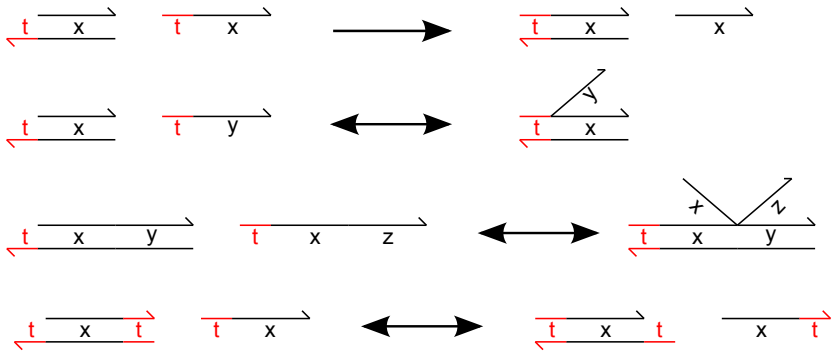



Figura 8: Migración y desplazamiento de cadenas mediada por puntos de apoyos o "toeholds". Adaptado de "A programming language for composable DNA circuits", De Phillips y Luca, 2009, Journal of The Royal Society Interface 6, 4 (2009), pp. S419-S436.

Una aplicación del modelo de desplazamiento de hebras es el diagnóstico oportuno de cáncer por Graugnard *et al.* (2010), quienes diseñaron una red química de ADN que acepta como entrada ácidos microribonucleicos (miARN), procesados por compuertas booleanas y cuya salida resulta en un gran número de cadenas de ADN que pueden ser detectadas fácilmente, con lo cual se amplifica las moléculas de miARN sin necesidad de utilizar PCR. Las moléculas de miARN son pequeñas cadenas no codificadoras de ARN entre 21 y 23 nucleótidos de longitud, las cuales actúan como reguladores de genes (aproximadamente 30 % de todos los genes y en la mayoría de las redes de regulación genética). Se ha demostrado que estas moléculas pueden funcionar como oncogenes o supresores de tumores, además de jugar papeles críticos en varios aspectos de progresión tumoral y en metastasis (Zhang *et al.*, 2008). Recientemente se comprobó que miARN puede detectarse en suero sanguíneo, lo que permite una detección no invasiva, aunque, este se encuentra en poca abundancia (Graugnard *et al.*, 2010). Actualmente, las tecnologías de diagnóstico requieren la extracción del ARN celular, producción de ADNc y amplificación por PCR para su detección. El sistema catalítico anterior en cambio, per-

mite la detección y amplificación de miARN específico, simplificando así el proceso de diagnóstico. Este sistema está compuesto de cuatro componentes distintos: el traductor encargado de la detección de las móleculas de miARN, la red catalítica transversal, la cual se encarga de la amplificación constante de la señal liberada por el traductor, y dos complejos reporteros encargados de liberar sondas luminosas ante la presencia de cadenas liberadas por el sistema catalítico.

# Capítulo 5. Diseño y optimización de complejos para el diagnóstico de mutaciones aberrantes

El diseño de sondas es importante para la realización de experimentos relacionados con ADN o ARN, principalmente en el diseño de microarreglos y cómputo basado con ADN. En este problema se busca generar secuencias de ADN o ARN independientes entre sí, donde por independencia se entiende como la mínima tendencia a hibridación cruzada y máxima diferencia entre ellas. Además, las sondas deben maximizar la tendencia a hibridar con la secuencia objetivo. Existen varios métodos y criterios para el diseño de sondas definidos en la literatura. Entre los criterios mayormente utilizados se encuentran el cálculo de energía libre entre secuencias (Rouillard *et al.*, 2003), el puntaje de correspondencia entre secuencias indebidas por medio de BLAST (Wang y Seed, 2003), la temperatura de fusión y la probabilidad de generación de estructuras secundarias como objetivo principal (Gordon y Sensen, 2004).

A diferencia de los esquemas anteriores, en el diseño de sondas para el diagnóstico de mutaciones aberrantes mediante genes computacionales, se busca una configuración de complejos de ADN para diagnóstico/tratamiento, tal que se maximice el número de verdaderos positivos, esto es, maximizar la detección correcta de moléculas de ADN o ARN mutadas, mientras se minimizan el número de falsos positivos o el número de detecciones incorrectas. Además, es necesario que el complejo Diagnóstico/Tratamiento sea termodinámicamente estable a 37°C, la temperatura promedio del cuerpo humano.

Una manera de medir la aptitud de un complejo es por medio del cálculo de energía libre de Gibbs entre señales de diagnóstico y tratamiento. En una reacción de hibridación de ácidos nucleicos, la energía libre de Gibbs mide la estabilidad del complejo generado, mientras $\triangle G^{\circ} \to -\infty$, mayor la estabilidad. De manera contraria, mientras $\triangle G^{\circ} \to \infty$,

el complejo se vuelve más inestable. La estabilidad de un complejo se correlaciona con el número de pares de bases que hibridan en una secuencia, por lo que a mayor número de pares de bases, mayor la estabilidad. Sin embargo, al aumentar el tamaño de un complejo, corre peligro de ser digerido por la célula huésped (Flintoft, 2010).

## 5.1. Modelo termodinámico

Basado en el modelo de genes computacionales por Martínez-Pérez *et al.* (2007), el problema se puede modelar como sigue. Sea $Am/AB'$ un complejo Diagnóstico/Tratamiento y $Am'$ la secuencia de ARNm que contiene la mutación a detectar. El proceso de detección se puede expresar mediante la reacción:

$$Am/AB' + Am' \underset{k_r}{\overset{k_f}{\rightleftharpoons}} Am/Am' + AB', \tag{26}$$

donde $k_f$ y $k_r$ son las constantes cinéticas de hibridación directa e inversa, calculadas con las ecuaciones (18) y (19), respectivamente. Como producto se obtiene una señal de tratamiento $AB'$ y un complejo ADN/ARN ($Am/Am'$). En esta reacción interesa que la constante de cinética de hibridación directa sea mucho mayor que la inversa, $k_f \gg k_r$, lo cual sugiere que de llevarse a cabo el desplazamiento de hebras, la reacción no será reversible.

El complejo Diagnóstico/Tratamiento también puede reaccionar con una secuencia de ARNm sano $Ah'$ de la siguiente manera:

$$Am/AB' + Ah' \underset{k_{er}}{\overset{k_{ef}}{\rightleftharpoons}} Am/Ah' + AB', \tag{27}$$

reacción que especifica un desplazamiento de cadenas errónea, es decir, un falso positivo.

Por esta razón, se busca que $k_{er} \gg k_{ef}$, para que en caso de que se lleve a cabo un desplazamiento de hebras, la reacción regrese a su estado inicial.

Observe que un complejo Diagnóstico/Tratamiento debe permanecer estable (no disociarse) para que las anteriores reacciones puedan ocurrir. La estabilidad de un complejo Diagnóstico/Tratamiento puede definirse como:

$$Am/AB' \underset{k_{as}}{\overset{k_{dis}}{\rightleftharpoons}} Am + AB', \tag{28}$$

donde $k_{dis}$ es la constante de disociación del complejo y $k_{as}$ la constante de asociación entre una sonda de diagnóstico, $Am$, y una sonda de tratamiento $AB'$. Lo que interesa en esta reacción es que los complejos se mantengan unidos y en caso de haber una separación espontánea, ésta regrese a su configuración más estable, por lo que se desea que $k_{as} \gg k_{dis}$.

De manera similar, una vez detectada la mutación objetivo, se requiere que el nuevo complejo ADN/ARN sea termodinámicamente estable, por lo que en la reacción

$$Am/Am' \underset{k_{ARNas}}{\overset{k_{ARNdis}}{\rightleftharpoons}} Am + Am', \tag{29}$$

se busca que el complejo se mantenga unido en todo momento y en caso de separarse este regrese a su estado híbrido, por lo que $k_{ARNas} \gg k_{ARNdis}$. La estabilidad del complejo ADN/ARN dentro de la célula es conveniente ya que dicho complejo sirve como sustrato de la endonucleasa celular RNasa H, lo que ayudaría a silenciar la expresión de la proteína disfuncional. Para calcular las constantes cinéticas se requiere entonces de conocer la energía libre del complejo ($\Delta G^{\circ}$) bajo condiciones fisiológicas estándar (pH = 7.0 y temperatura de incubación a 37°C, que es la temperatura promedio de la célula). Además, es necesario que las sondas se diseñen de tal manera que su configuración no

requiera de una gran cantidad de nucleótidos, con la finalidad de minimizar el riesgo de generar estructuras secundarias no previstas, minimizar el costo de síntesis y tratar de proteger las sondas de endonucleasas celulares. Sin embargo, esto genera un conflicto, ya que a mayor longitud, mayor el número de nucleótidos apareados y por lo tanto, mejor su estabilidad.

En resumen, las funciones a considerar son:

Constantes de hibridación directa e inversa $k_f$ y $k_r$ en una detección correcta (verdadero positivo).

Constantes de hibridación directa e inversa $k_{ef}$ y $k_{er}$ en una detección errónea (falso positivo).

Estabilidad del complejo Diagnóstico/Tratamiento.

Estabilidad del complejo ADN/ARN.

## 5.2. Problema de optimización

Dadas dos secuencias $Am'$ y $Ah'$, la secuencia objetivo y contraobjetivo respectivamente, se busca una configuración en el complejo de Diagnóstico/Tratamiento tal que maximice la detección de la secuencia objetivo $Am'$, la cual representa una secuencia de ARNm mutado, mientras se minimiza el número de detecciones erróneas debidas a un desplazamiento de cadenas con la secuencia $Ah'$ (una secuencia de ARNm sano).

Formalmente, el problema se puede definir como:

$$\begin{aligned}
&\text{Minimizar } f_i(X),\ i \in \{\Delta G^{\circ}_{ComplejoDT}, \Delta G^{\circ}_{ADN/ARN}, k_{rDT}, k_r, k_{er}, longitud\},\\
&\text{maximizar } f_j(X),\ j \in \{\Delta G^{\circ}_{ADN/ARNh}, k_{fDT}, k_f, k_{ef}\},\\
&\quad \text{sujeto a } Tm = 37^{\circ}\text{C, pH} = 7.0, \quad \text{y}\\
&\qquad \Delta G^{\circ}_{ADN/hARN} < \Delta G^{\circ}_{ComplejoDT} < \Delta G^{\circ}_{ADN/ARN},
\end{aligned} \tag{30}$$

donde $\Delta G^{\circ}_{ComplejoDT}$ es la energía libre de Gibbs del complejo Dx/Tx (Ecuación (3)), $k_{fDT}$ y $k_{rDT}$ las constantes de cinética de asociación y disociación del complejo Dx/Tx (Ecuaciones (18) y (19)), $\Delta G^{\circ}_{ADN/ARN}$ la energía libre de Gibbs del complejo ADN/ARNm mutado en una detección correcta (Ecuación (3)), $k_f$ y $k_r$ son las constantes de cinética de hibridación de avance y retroceso del complejo ADN/ARNm mutado (Ecuaciones (18) y (19)), $\Delta G^{\circ}_{ADN/ARNh}$ es la energía libre de Gibbs del complejo ADN/ARNm sano en una detección erronéa (Ecuación (3)),, $k_{ef}$ y $k_{er}$ son constantes de cinética de hibridación de avance y retroceso del complejo ADN/ARNm sano (Ecuaciones (18) y (19)).

## 5.3. Caso de estudio: detección de la mutación $\Delta F508$ del gen de la fibrosis quística.

Hasta el momento se ha definido un modelo general para la optimización de sondas encargadas de diagnosticar mutaciones aberrantes, el cual es independiente del tipo de mutación presente. Sin embargo, para ejemplificar una hipotética aplicación, este trabajo se enfocará en el diagnóstico y tratamiento teórico de la fibrosis quística originada por la mutación $\Delta F508$. Como se recordará, esta mutación se caracteriza por la deleción de un codón (`CTT`), el cual provoca la pérdida de un residuo de fenilalanina en la

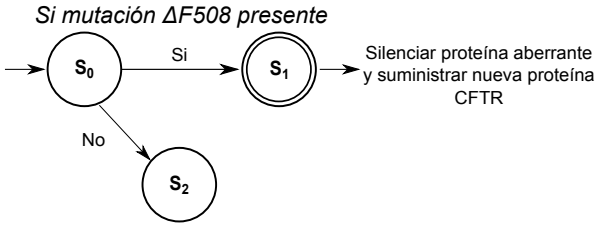



Figura 9: Autómata determinístico encargado de la detección y terapia de la mutación $\Delta F508$.

posición 508 de la proteína. De acuerdo a la Ecuación 25, el diagnóstico y tratamiento de esta enfermedad podría realizarse mediante la siguiente regla:

$$\begin{gathered} Si \text{ el codón } \texttt{CTT} \text{ no está presente en la posición 508} \\ entonces \text{ silenciar proteína aberrante y generar nueva proteína FQ.} \end{gathered} \tag{31}$$

Esta regla puede ser implementada utilizando un autómata de tres estados y un símbolo (Figura 9), en donde la única manera de transitar de un estado inicial $S_0$ a un estado final de aceptación $S_1$ (suministro de terapia) es con la presencia de la mutación $\Delta F508$. Este autómata se puede modelar con un gen computacional similar al de la Figura 7.

En el modelo original, dicho autómata se implementó para el diagnósico y tratamiento de mutaciones puntuales, fungiendo éstos como marcadores moleculares para el proceso de detección. Sin embargo, para el caso de la fibrosis quística la mutación estudiada se presenta mediante la eliminación de pares de bases, lo que hace imposible su uso como marcador molecular, por lo que es necesario adaptar el complejo diagnóstico/tratamiento del gen computacional.

Considerando esta restricción, la detección se realiza ahora de la siguiente manera: sea $ARNm\,sano = 3'\text{-}B_{i-p}\ldots B_{i-2}B_{i-1}B_iB_{i+1}B_{i+2}\ldots B_{i+q}\text{-}5'$ una secuencia de ARN de $q+p+1$ bases de longitud, donde $p < i < q$. Suponga que a la cadena $ARNm\,sano$ le ocurre una mutación consistente de la deleción de una tripleta de nu-

cleótidos en las posiciones $B_iB_{i+1}B_{i+2}$. Se define entonces la cadena mutada resultante como $ARNm\,mutado = 3'\text{-}B_{i-p}\ldots B_{i-2}B_{i-1}B_{i+3}\ldots B_{i+q}\text{-}5'$, donde los nucleótidos vecinos de la tripleta eliminada (e.g. $B_{i-3}B_{i-2}B_{i-1}B_{i+3}B_{i+4}B_{i+5}$) podrían servir como marcador molecular de la mutación. La señal de diagnóstico de un complejo Dx/Tx se puede diseñar con una secuencia Dx $= 5'\text{-}\overline{B}_{i-p}\ldots\overline{B}_{i-2}\overline{B}_{i-1}\overline{B}_{i+3}\ldots\overline{B}_{i+q}\text{-}3'$, por lo que Dx es totalmente complementaria con $ARNm\,mutado$ y parcialmente complementaria con $ARNm\,sano$. Idealmente, la señal de tratamiento Tx debería diseñarse de tal manera que $\Delta G^{\circ}_{ADN/ARN} < \Delta G^{\circ}_{ComplejoDT} < \Delta G^{\circ}_{ADN/ARNh}$.

Observe que la primera desigualdad garantizaría la detección de verdaderos positivos, mientras que la segunda desigualdad evitaría la detección de falsos positivos. Sin embargo, el cumplimiento de la última desigualdad depende de la elección correcta de los nucleótidos que pertenecerán a la sección no complementaria del complejo Dx/Tx.

Por ejemplo, en el gen de la fibrosis quística, el correspondiente transcrito tiene la secuencia $3' - \texttt{UUU AUA }\underline{\texttt{GUA}}\texttt{ GAA }\underline{\texttt{ACC}}\texttt{ ACA AAG} - 5'$. Cuando la mutación $\Delta F508$ ocurre, la tripleta `GAA` (en negrita) se pierde, uniendo como consecuencia los codones vecinos $3' - \texttt{UUU AUA }\underline{\texttt{GUAACC}}\texttt{ ACA AAG} - 5'$. La señal de diagnóstico del complejo Dx/Tx puede diseñarse como una cadena sencilla de ADN, la cual es completamente complementaria al ARNm mutado: $5' - \texttt{AAA TAT }\underline{\texttt{CATTGG}}\texttt{ TGT TTC} - 3'$, y parcialmente complemenario al ARNm sano. Por otro lado, la señal de tratamiento tiene que diseñarse de tal manera que el complejo Dx/Tx solamente se separe ante la presencia de moléculas de ARNm enfermas. Se define la señal de tratamiento como $3'-\underline{\texttt{TTT}}\ \underline{\texttt{ATA}}\ \underline{\texttt{G}}\texttt{GGG}\underline{\texttt{CC}}\ \underline{\texttt{ACA}}\ \underline{\texttt{AAG}}-5'$, la cual es parcialmente complementaria a la señal Dx (sección subrayada), de esta manera, la detección se realiza mediante el codon `ATT` del complejo Dx, fungiendo como marcador molecular. La configuración del complejo así como el mecanismo de detección se puede apreciar en la Figura 10.

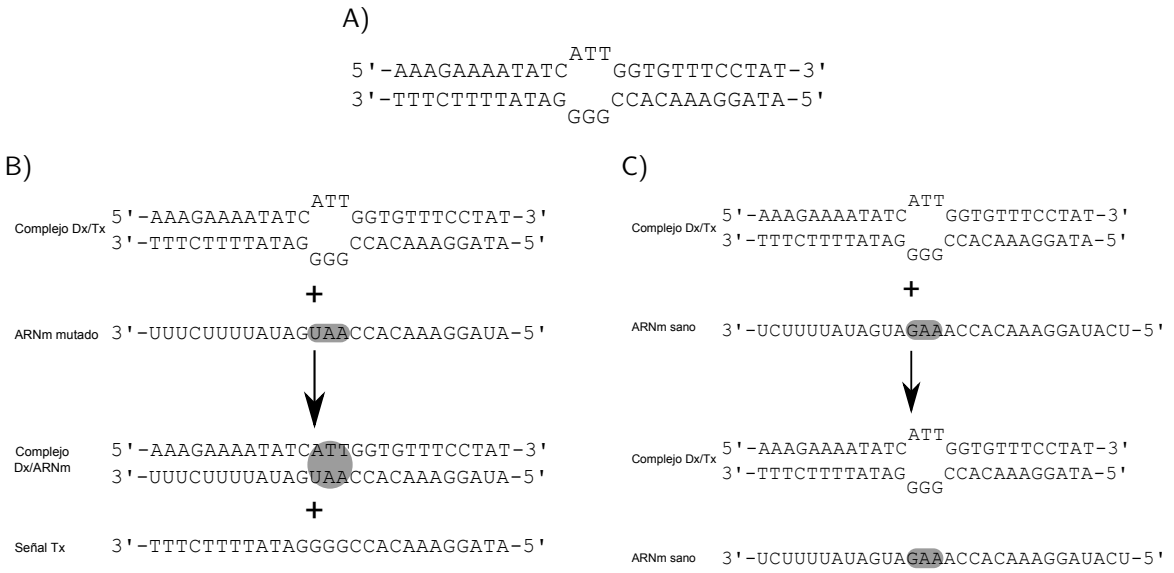



Figura 10: Configuración del complejo Dx/Tx y mecanismo de detección para la mutación $\Delta F508$ de la fibrosis quística. A) Complejo Dx/Tx . B) Diagnóstico positivo correcto impulsado por la mayor estabilidad del complejo ADN/ARN gracias a la asociación de las bases entre la señal de diagnóstico y el marcador molecular (resaltado en gris). C) Diagnóstico correcto al no existir la mutación

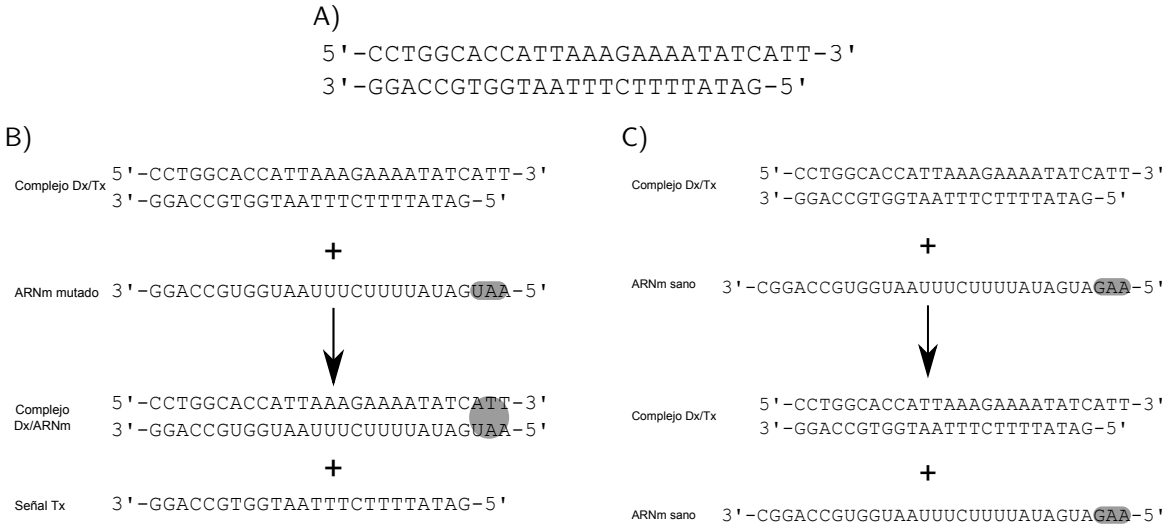



Figura 11: Configuración por nucleótidos sobresalidos 3' o derecha (señal Dx) del complejo Dx/Tx. A) Complejo Dx/Tx. B) Diagnóstico positivo correcto impulsado por la mayor estabilidad del complejo ADN/ARN gracias a la asociación de las bases entre la señal de diagnóstico y el marcador molecular (resaltado en gris). C) Diagnóstico correcto al no existir la mutación


Figura 12: Configuración por nucleótidos sobresalidos 5' o izquierda (señal Dx) del complejo Dx/Tx. A) Complejo Dx/Tx . B) Diagnóstico positivo correcto impulsado por la mayor estabilidad del complejo ADN/ARN gracias a la asociación de las bases entre la señal de diagnóstico y el marcador molecular (resaltado en gris). C) Diagnóstico correcto al no existir la mutación

En el modelo original de genes computacionales, se contempla únicamente complejos Dx/Tx con estructuras secundarias tipo bucle interior. Para este trabajo se amplió la capacidad de representar la mutación mediante dos nuevas estructuras secundarias, estas son nucleótidos sobresalientes $3'$ (Figura 11) y nucleótidos sobresalientes $5'$ (Figura 12) respecto a la señal Dx.

## 5.4. Optimización multi-objetivo de complejos Dx/Tx

Como se menciona en el trabajo de Martínez-Pérez *et al.* (2007), uno de los factores clave en el desempeño de un gen computacional es el diseño óptimo de complejos diagnóstico/tratamiento. Entre las variables se encuentran la longitud de las sondas y las estructuras secundarias de las mismas. Sin embargo, diseñar experimentalmente las sondas en laboratorio conlleva tiempo y recursos, por lo que se propone realizar un modelo de optimización computacional, utilizando los parámetros ya conocidos para el

cálculo de energía libre en complejos de ADN y ARN (Xia *et al.*, 1998; SantaLucia y Hicks, 2004). Entre las diferentes heurísticas existentes para optimización se optó por utilizar algoritmos evolutivos multi-objetivo, en específico el algoritmo NSGA-II desarrollado por Deb *et al.* (2002). A continuación se explican los diferentes componentes desarrollados, necesarios para resolver el problema de optimización de complejos.

## 5.5. Representación del individuo

Los complejos ADN/ADN y ADN/ARN son estructuras tridimensionales que se diferencían en la configuración de su hélice, el número de nucleótidos no complementarios y el tipo de estructura secundaria que generan (ya sean bucles interiores, bultos, extremos sobresalientes, etc.), por lo que intentar representar tales configuraciones se complica. Buscando simplificar el problema, se decidió modelar los complejos considerando estructuras secundarias sencillas, entre las cuales encontramos:

Bucles interiores.

Bultos superiores e inferiores

Extremos 3' y 5' con nucleótidos sobresalientes.

Regiones complementarias.

Estas configuraciones deben representarse de tal manera que los operadores de cruzamiento y mutación del algoritmo genético no requieran de un gran tiempo de cómputo. Phillips y Luca (2009) diseñaron un lenguaje capaz de representar este tipo de estructuras de manera lineal, lo cual posibilita la esquematización de una representación sencilla en el algoritmo evolutivo.

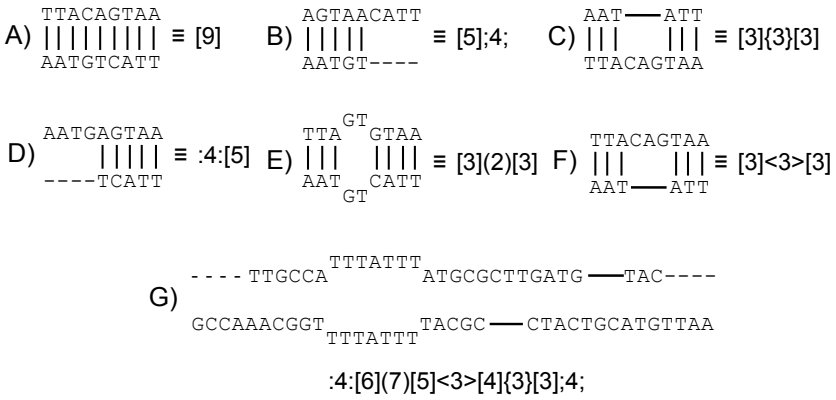



Figura 13: Representación de estructuras secundarias. A) Región de nucleótidos complementarios de tamaño 9. B) Extremo derecho (5′) con 4 nucleótidos sobresalientes. C) Bulto inferior de longitud 3. D) Extremo izquierdo (3′) con 4 nucleótidos sobresalientes. E) Bucle interior de tamaño 2. F) Bulto superior de longitud 3. G) Ejemplo de complejo diagnóstico/tratamiento.

Basándose en este modelo, se define la representación de la información en el cromosoma de un individuo, en el que básicamente, un individuo se representa como una cadena lineal de caracteres de tamaño variable, el cual debe cumplir con las siguientes restricciones:

Las secciones de nucleótidos sobresalientes solo pueden estar en el extremo 3′ y 5′ de la señal Dx del complejo diagnóstico/tratamiento.

Después de cualquier estructura secundaria no complementaria, inmediatamente le sigue una región complementaria; y viceversa.

Cualquier tipo de bulto o bucle tiene que estar posicionado entre dos secciones complementarias.

Una región complementaria no puede tener como vecina otra región complementaria.

El individuo debe poseer una región que permita el desplazamiento de hebras. Para lograr este objetivo, una configuración debe tener al menos un bucle interior

o extremos sobresalientes 3' o 5' que contengan la secuencia que permita discernir entre moléculas de ARNm sano y ARNm mutado.

De acuerdo a las restricciones dadas, se define la gramática para la representación del cromosoma de un individuo como:

| | | |
|---|---|---|
| cromosoma | = | <extremo sobresaliente de inicio> <región interna> \| <región interna> |
| <región interna> | = | <región complementaria> <estructura secundaria> <región interna> \| <fin de cromosoma> |
| <fin de cromosoma> | = | <región complementaria> \| <región complementaria> <extremo sobresaliente final> |
| <extremo sobresaliente de inicio> | = | ':' <número> ':' |
| <región complementaria> | = | '[' <número> ']' |
| <estructura secundaria> | = | '(' <número> ')' \| '<' <número> '>' \| '{' <número> '}' |
| <extremo sobresaliente final> | = | ';' <número> ';' |
| <número> | = | $n \in \mathbb{N}$ |

Además de la representación de estructuras, el individuo necesita contener información acerca de las secuencias de ADN que contendrá el complejo. Esto es importante ya que la sonda de diagnóstico Dx tiene que ser totalmente complementaria al ARNm mutado. Dada una secuencia plantilla $\Theta \in \Lambda$, donde $\Lambda \in \{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}\}^*$, se puede especificar una secuencia $\theta \subseteq \Theta$ a utilizar en el complejo por medio de índices $\alpha, \omega$ tal que

$\alpha < \omega \leq |\Theta|$, donde $|\Theta|$ es la longitud de $\Theta$.

Observe que la señal Tx es la encargada de mantener la estructura deseada del complejo Dx/Tx. Por consiguiente, la secuencia de la misma estará especificada por las regiones complementarias en el complejo Dx/Tx.

Otro dato necesario para la representación a utilizar, es un índice que describa la posición de la subestructura secundaria encargada de la detección de la mutación con respecto al cromosoma del individuo, $\gamma$, el cual se utilizará en los operadores de cruce y mutación. Por lo que el individuo a utilizar en el AG se define como:

$$\begin{aligned} \texttt{individuo} &= < \alpha >< \omega >< \gamma >< \texttt{cromosoma} > \\ < \alpha > &= \alpha \in \mathbb{N} \mid \alpha \geq 0 \\ < \omega > &= \omega \in \mathbb{N} \mid \omega < n \leq |\Theta| \\ < \gamma > &= \gamma \in \mathbb{N} \mid \gamma < |\texttt{cromosoma}| \end{aligned}$$

de esta manera, la estructura del complejo Dx/Tx se encuentra representada mediante el cromosoma del individuo, mientras que las secuencias de nucleótidos contenidos en el complejo es obtenido por medio de los índices de inicio$\alpha$ y fin $\omega$, los cuales corresponden a la secuencia plantilla $\Theta$.

## 5.6. Diseño del algoritmo genético

### 5.6.1. Inicialización de la población

Antes de inicializar la población, el algoritmo recibe las plantillas correspondientes a las secuencias de ARNm mutado y sano. Tomando como plantilla estas secuencias, la población de padres se inicializa de manera aleatoria utilizando la gramática previamente definida. Una vez construida la población inicial, los individuos son ingresados en

las funciones objetivo establecidas con anterioridad (Ecuación 30), asignando la aptitud correspondiente a la población.

### 5.6.2. Selección de padres

En cada generación, la población a evaluar se clasifica conforme el criterio de no dominancia otorgada por sus funciones objetivo. La factibilidad de las soluciones no dominadas se asegura al utilizar selección por torneo restringido. En este método, al compararse un par de individuos, (1) las soluciones factibles son preferidas ante aquellas no factibles; (2) entre dos soluciones no factibles, la menos penalizada es seleccionada y; (3) entre dos soluciones factibles, la no dominada es preferida. En caso de que las soluciones coincidan en el mismo frente no dominado, la solución con mayor distancia de amontonamiento es seleccionada.

### 5.6.3. Selección de sobrevivientes

Para seleccionar a los sobrevivientes, la generación de hijos se mezcla con la generación previa de padres. Cada individuo en la población extendida ($R$) es evaluado y ordenado basado en su no dominancia. Después se efectúa una selección por torneo restringido en $R$ para generar una nueva población elitista de padres $P'$. Este proceso es repetido hasta que se haya alcanzado el número máximo de generaciones.

## 5.7. Operadores genéticos

Debido a la representación utilizada, es necesario definir operadores especiales que acepten tal representación y respeten características claves en los complejos, tales como la posición de la estructura no complementaria encargada de la detección de la mutación,

la gramática del individuo y los índices que especifican la secuencia de la señal Dx. A continuación se describen operadores de variación utilizados en el algoritmo evolutivo.

### 5.7.1. Operadores de cruzamiento

Para este trabajo se desarrollaron dos operadores de cruzamiento análogos a operadores comúnmente utilizados en cromosomas binarios. El primer operador se basa en el operador de cruzamiento de un solo punto propuesto por Holland (1992), mientras que el segundo operador es semejante al operador de cruzamiento uniforme (Mühlenbein, 1997).

#### 5.7.1.1. Operador de cruzamiento de un solo punto

El esquema básico es el siguiente: dados dos padres, este operador genera dos nuevos hijos en dos fases: (1) alineando ambos padres mediante las subestructuras $\gamma$ y (2) intercambiando la información a partir de dicha subestructura (Figura 14). Cabe mencionar que este cruzamiento puede variar según la estructura que contenga la mutación en el complejo. Por ejemplo, si la mutación se encuentra localizada en una estructura con nucleótidos sobresalientes en el extremo $5'$ (lado izquierdo de la señal Dx), el cruce se realiza de izquierda a derecha, esto es, de la subestructura $\gamma + 1$ hasta la subestructura $n$ (Figura 14C). Individuos cuya mutación se encuentra en nucleótidos sobresalientes del extremo $3'$ (lado derecho de la sonda Dx) se cruza de derecha a izquierda, de la subestructura $\gamma - 1$ hasta la subestructura 1 (Figura 14B). Por otro lado, las representaciones por bucle interior se pueden cruzar por ambos lados, esto es, de la subestructura 1 hasta la subestructura $\gamma$ de un padre aleatorio y la información de la subestructura $\gamma + 1$ hasta la subestructura $n$ del otro, y viceversa (Figura 14A). Un caso especial de cruzamiento es cuando el marcador de mutación de un padre se

encuentra en el extremo $5'$ y en el extremo $3'$ del otro , entonces, el cruzamiento se lleva a cabo copiando la información de manera inversa de la subestructura $\gamma_1 - 1$ hasta la subestructura 1 del primer padre con extremo sobresaliente $3'$ y la información de la subestructura $\gamma_2 + 1$ hasta la subestructura $n$ del segundo padre (con extremo sobresaliente $5'$), formandose un nuevo hijo. De manera análoga, el segundo hijo es formado con la información de la subestructura $\gamma_2 + 1$ hasta la subestructura $n$ del padre con extremo sobresaliente $3'$ y la información restante del otro padre; y de ser necesario, son reparadas las configuraciones erróneas (Figura 14D). De igual manera, el cruzamiento entre diferentes representaciones de complejos se lleva a cabo respetando las reglas anteriores. Al finalizar el intercambio de la información, se recalculan los índices $\alpha_i, \omega_i, \gamma_i$ de los hijos, ya que al cruzar ambas estructuras con sus respectivas secuencias se ven modificadas.

#### 5.7.1.2. Operador de cruzamiento uniforme

El esquema de este operador es el siguiente: dado dos padres, este operador genera dos nuevos hijos mediante dos fases; (1) los padres se alinean mediante las subestructuras $\gamma$ y después (2) se intercambia la información de ambos padres a partir de esta subestructura. Sin embargo, a diferencia del operador anterior, este operador cruza una por una las subestructuras (con igual probabilidad) que se encuentren alineados en un par de individuos. La forma en que el cruce se realiza depende del tipo de estructura que contenga la mutación en el complejo (Figura 15). Padres con representación por bucle interior se cruzan completamente, utilizando la información desde la subestructura inicio hasta la subestructura $\gamma$, y la información de la subestructura $\gamma + 1$ hasta la subestructura $n$, copiando subestructura por subestructura de manera aleatoria a los hijos (Figura 15A). Padres con extremos sobresalientes se cruzan solamente del lado iz-

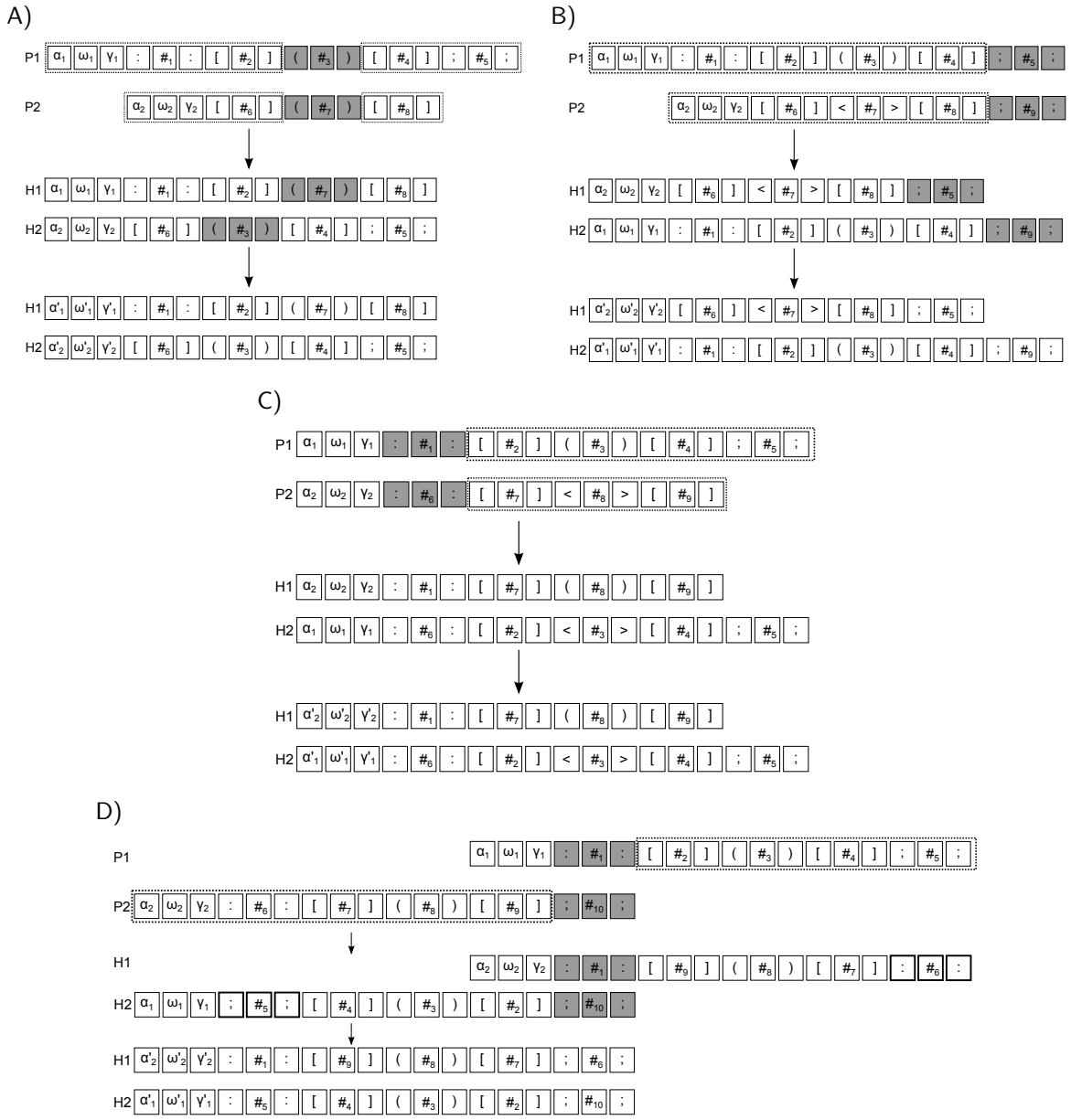



Figura 14: Esquemas de cruzamiento de un solo punto. Dados dos padres, la información a intercambiar entre ellos (recuadro con líneas discontinuas) dependerá de la subestructura que contiene al marcador de la mutación objetivo (resaltado en gris). A) Padres con bucle interior, B) padres con extremo $3'$ sobresaliente, C) padres con extremo $5'$ sobresaliente y D) padres cuyo marcador de mutación se localiza en diferentes extremos de la señal Dx. En todos los casos se recalculan los índices $\alpha_i$, $\omega_i$ y $\gamma_i$, y se verifican las subestructuras no válidas.

quierdo ($5'$) y derecho ($3'$), respectivamente. Esto es, la información de la subestructura de inicio (1) hasta la subestructura $\gamma$ es copiada de los padres a los hijos de manera uniforme en una representación sobresaliente $3'$ (Figura 15B), y de la subestructura $\gamma$ hasta la subestructura $n$ en representaciones sobresalientes por lado izquierdo (Figura 15C). Un caso especial es cuando el marcador de mutación de un padre se encuentra en el extremo $5'$ ($\gamma_1$) y en el extremo $3'$ del otro ($\gamma_2$). En este caso, se realiza un alineamiento virtual entre $(\gamma_{1+1}, \gamma_{2-1})$, $(\gamma_{1+2}, \gamma_{2-2})$, $(\gamma_{1+3}, \gamma_{2-3})$, etc., mientras que el cruzamiento se lleva a cabo de manera uniforme copiando la información contenida (de manera aleatoria) de cada par de subestructuras alineadas a los hijos (Figura 15D). De igual manera, el cruzamiento entre diferentes representaciones de complejos se lleva a cabo respetando las reglas anteriores. Otro punto a considerar es el tamaño de los individuos. Si dos padres son del mismo tamaño, el cruzamiento se realiza de manera uniforme, intercambiándose subestructuras entre padres con la misma probabilidad. En cambio, si uno de los dos padres es más grande, el cruzamiento se llevará a cabo de manera aleatoria hasta el final del padre de menor longitud, mientras la configuración restante es heredada aleatoriamente a cualquiera de los hijos. Posteriormente, se recalculan los índices de los hijos, además de revisar si su configuración es válida, esto es, que cumplan las restricciones previamente establecidas, por lo que es necesario realizar un recorrimiento lineal para revisar y reparar cada uno de los hijos generados por este operador.

### 5.7.2. Operadores de mutación

En total se desarrollaron tres diferentes operadores de mutación. El primero es una adaptación del operador de inversión para representación binaria. El segundo se basa en el operador de deleción y, por último, el tercer operador es una combinación entre los operadores de intercambio, deleción e inserción.

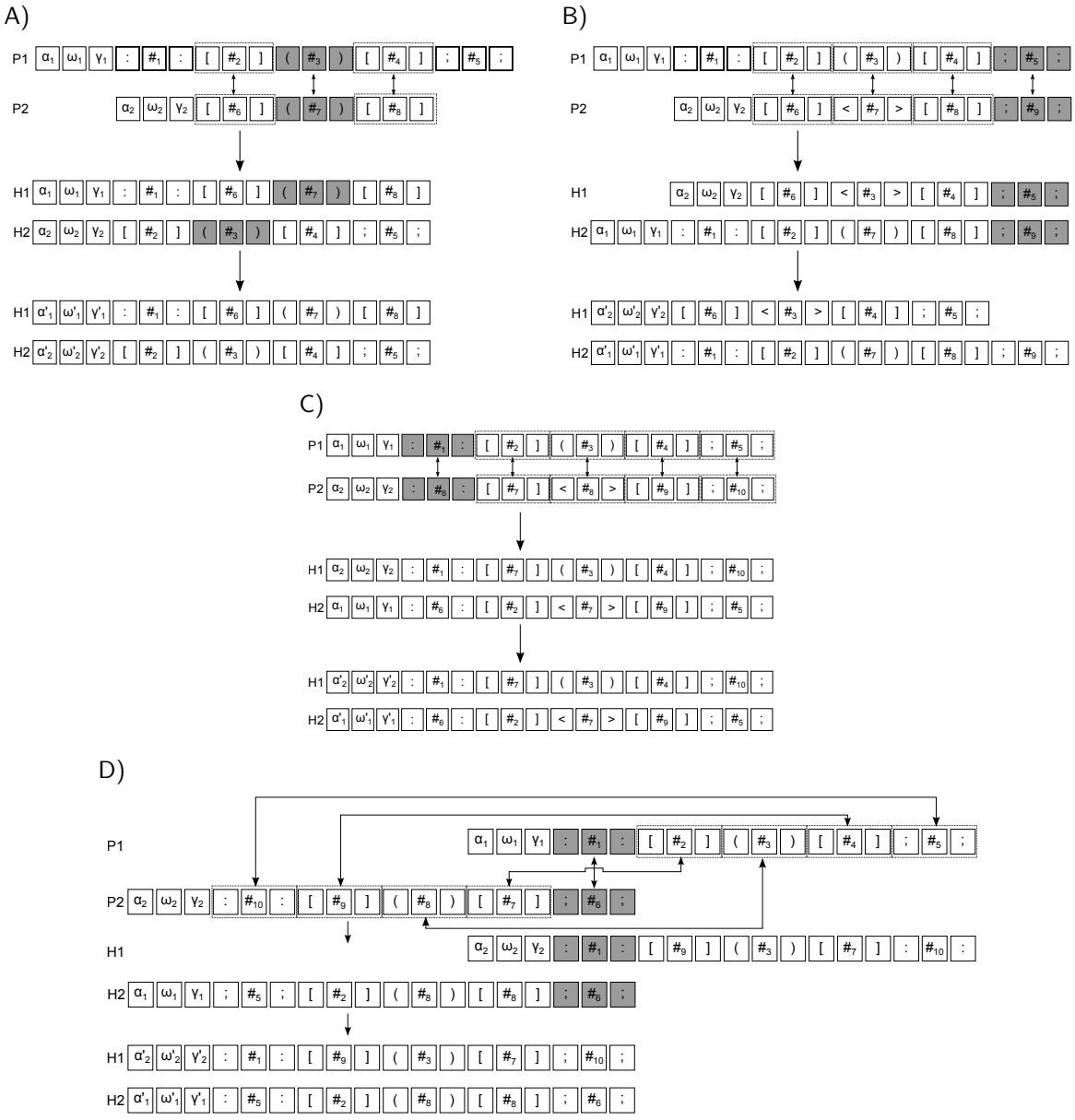



Figura 15: Esquema de cruzamiento uniforme. Dados dos padres, la información a intercambiar entre ellos (recuadro con líneas discontinuas) dependerá de la subestructura que contiene el marcador de la mutación objetivo (resaltado en gris). A) Padres con bucle interior, B) padres con extremo $3'$ sobresaliente, C) padres con extremo $5'$ sobresaliente y D) padre 1 con extremo $3'$ sobresaliente y padre 2 con extremo $5'$ sobresaliente. En todos los casos se recalculan los índices $\alpha_i$, $\omega_i$ y $\gamma_i$, y se verifican las subestructuras no válidas.

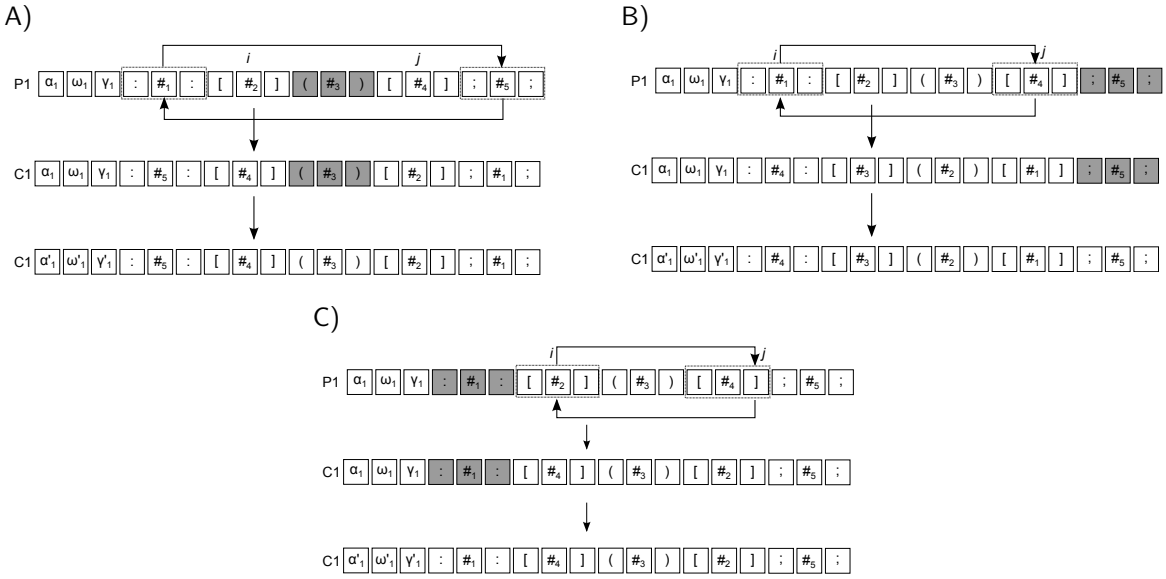



Figura 16: Operador de mutación de inversión. Dadas dos subestructuras $i$ y $j$ aleatorias (resaltado en líneas no continuas), se realiza una inversión en el número de nucleótidos contenido entre las subestructuras $i$ y $j$, de acuerdo a la subestructura que contiene el marcador de mutación (resaltado en gris). A) Individuo con bucle interno, B) individuo con configuración sobresaliente derecha, C) individuo con configuración sobresaliente izquierda. Al finalizar el proceso se recalculan los índices $\alpha_i$, $\omega_i$ y $\gamma_i$.

#### 5.7.2.1. Mutación por inversión

Este operador funciona seleccionando de manera aleatoria dos subestructuras $(i, j)$ en el individuo e invierte el orden del subconjunto generado entre las posiciones (Figura 16). La inversión se realiza en los valores numéricos contenidos en las subestructuras del complejo (< número >) que representan el número de nucleótidos que contiene cada uno de ellas. La selección de los puntos depende de la configuración del individuo. Mientras en un bucle (Figura 16A)) la selección de puntos se puede realizar antes o después del marcador de mutación $\gamma$, esto es, $(i, j) < \gamma$ o $(i, j) > \gamma$, en una configuración de nucleótidos sobresalientes los puntos se seleccionan antes de $\gamma$, $(i, j) < \gamma$, cuando es del extremo $3'$ (Figura 16B)), ó después de $\gamma$, $(i, j) > \gamma$, cuando es por el extremo $5'$ (Figura 16C)).

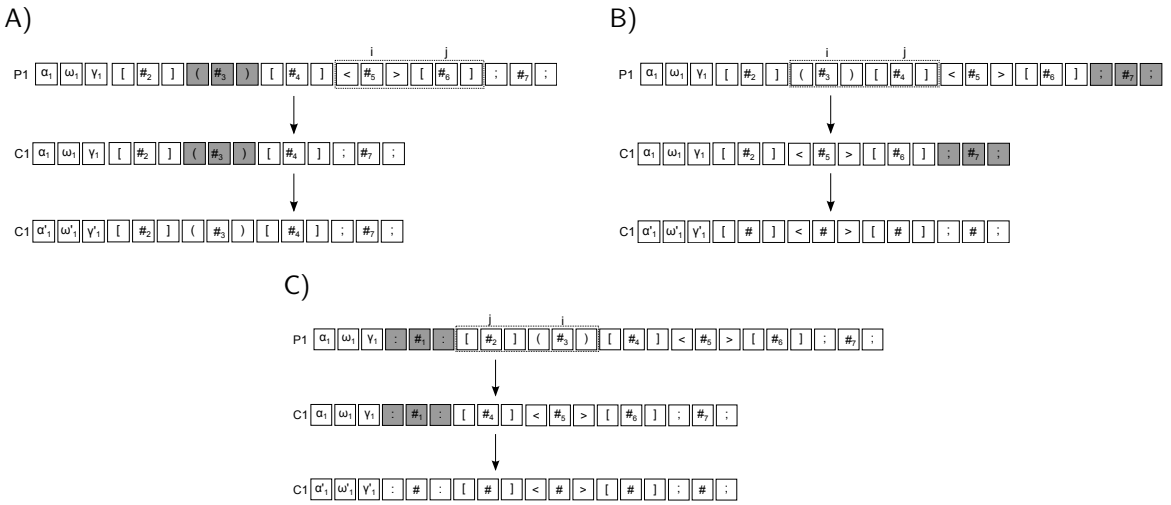



Figura 17: Operador de mutación por deleción. Dada una subestructura no complementaria aleatoria $i$, y una subestructura vecina $j$, tal que $i, j < \gamma$ ó $i, j > \gamma$, dependiendo del marcador de mutación $\gamma$ del individuo (resaltado en gris), se eliminan las subestructuras seleccionadas por los índices $i, j$. A) Individuo con bucle interno, B) individuo con configuración sobresaliente derecha, C) individuo con configuración sobresaliente izquierda.

#### 5.7.2.2. Mutación por deleción

Este operador funciona seleccionando de manera aleatoria una subestructura no complementaria $i$ de un individuo y una subestructura vecina $j$, tal que $j < i < \gamma$ ó $\gamma < i < j$. La subsecuencia contenida entre $i, j$ es eliminada recalculándose los índices pertinentes. Al igual que el operador anterior, la selección de los puntos está restringida por la subestructura que contiene la mutación en un individuo ($\gamma$). Este operador se aprecia en la Figura 17.

#### 5.7.2.3. Mutación por complemento

Este operador, mostrado en la Figura 18, selecciona de manera aleatoria una subestructura $i$, tal que $(i < \gamma)$ ó $(i > \gamma)$, dependiendo de la configuración con la que se representa la mutación en el individuo, y una subestructura vecina $j < i$ ó $j > i$, para después eliminar un número aleatorio de nucleótidos de la subestructura $j$ y agregar el

mismo número de nucleótidos a la subestrucura vecina $i$. Como se puede observar, este operador no necesita recalcular los índices $\alpha_i, \omega_i, \gamma_i$ en el individuo, ya que la cantidad total de nucleótidos no se modifica.

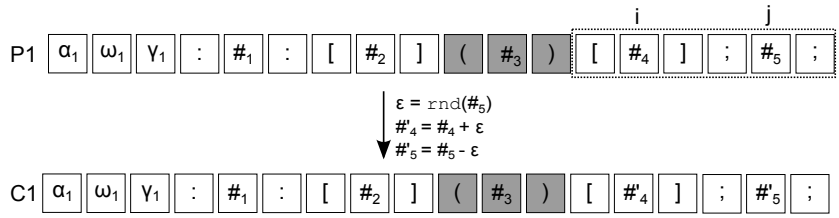



Figura 18: Operador de mutación por complemento. Dada una subestructura aleatoria $i$ tal que $(i < \gamma)$ ó $(i > \gamma)$, dependiendo de la configuración, y una subestructura $j$, el operador elimina un número aleatorio de nucleótidos de la subestructura $j$ y los agrega a la subestructura $i$.

## 5.8. Plataforma modular para optimización por medio de meta-heurísticas OPT4J

El sistema OPT4J es una plataforma para la aplicación de algoritmos meta heurísticos de optimización para problemas arbitrarios (Lukasiewycz *et al.*, 2011). OPT4J incluye algoritmos evolutivos multi-objetivo (SPEA2 y NSGA2), Evolución Diferencial multi-objetivo, Optimización por cúmulo de particulas multi-objetivo (PSO) y Recocido simulado mono-objetivo. Desarrollado por el Departamento de Ciencias de la Computación de la universidad de Erlangen-Nuremberg, Alemania, la principal ventaja de esta plataforma es su programación modular, lo cual permite que el problema pueda simplificarse en diferentes módulos para después ser codificado. Siguiendo este esquema, es posible utilizar cualquiera de los algoritmos mencionados para iniciar la búsqueda de soluciones. Otra ventaja que ofrece esta plataforma es su interfaz visual, la cual facilita el trabajo al optimizador al mostrarle resultados en ventanas y tablas, además de generar gráficas sencillas donde se muestra el comportamiento de cada una de las

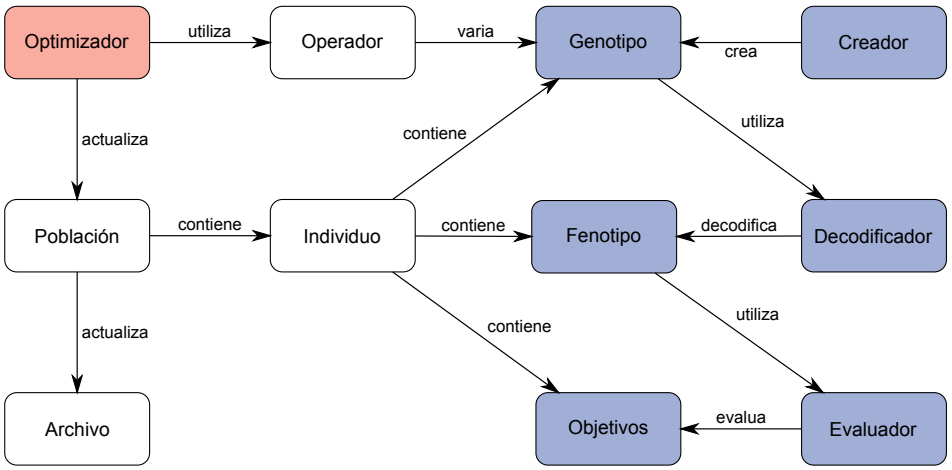



Figura 19: Esquema de interfaces y clases importantes en OPT4j. Adaptado de "Opt4J - A Modular Framework for Meta-heuristic Optimization", De Lukasiewycz *et al.*, 2011, Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 2011), 2011, 1723-1730.

funciones objetivo, incluyendo el trazado de frentes de Pareto en dos dimensiones.

Generalmente, OPT4J permite encontrar soluciones a problemas de optimización genéricos, esto es, problemas en donde se utilice una representación de individuos que sea común (e.g. representación binaria). En OPT4J cada individuo se conforma por: un genotipo que representa la codificación del problema (donde se llevará a cabo el cruzamiento y la mutación), un fenotipo que almacena las características físicas del individuo, y un evaluador que recibe el fenotipo del individuo y lo ingresa en las funciones objetivo para su evaluación (Figura 19).

Para la realización de este trabajo, se implementó el genotipo, fenotipo, las funciones de evaluación, así como los operadores de variación anteriormente descritos. Se modificó además la plataforma por razones de compatibilidad con la representación del individuo y se desarrolló un generador aleatorio de individuos con dicha representación. Para el problema a resolver no es necesaria una decodificación, por lo que el genotipo y el fenotipo del individuo es el mismo.

## 5.9. Configuración de corridas y resultados

En todo problema de optimización heurístico es necesario calibrar el algoritmo con la finalidad de encontrar una configuración de parámetros que genere buenos resultados. En un algoritmo genético, los parámetros comúnmente utilizados son el número de generaciones, el tamaño de población, el número de padres e hijos por generación, y las probabilidades de cruce y mutación. Además, en caso de contar con diferentes operadores de cruzamiento, mutación, así como representaciones diferentes para el problema, también es necesario su calibración. Para el problema en cuestión, se realizaron diferentes corridas permutando el número de generaciones, las probabilidades y operadores de cruce y mutación (Tabla 4).

Tabla 4: Parámetros a permutar en el optimizador.

| Parámetro | Valores a permutar | |
|---|---|---|
| Probabilidad de cruce | 0.95 | Cruce de un punto y uniforme |
| | 0.90 | |
| | 0.85 | |
| Probabilidad de mutación | 0.01 | Inversión, deleción y complemento |
| | 0.05 | |
| | 0.1 | |
| Número de generaciones / Tamaño de población | 500/1,000 | |
| | 250/2,000 | |
| | 150/4,000 | |

En total se probaron 162 configuraciones diferentes ($2 \times 3 \times 3 \times 3 \times 3$). Además, cada combinación de parámetros se ejecutó 10 veces, haciendo un total de 1,620 experimentos. Durante una corrida en OPT4J se generó un archivo donde se almacena el frente de Pareto no dominado por generación. Para facilitar la obtención de datos y su posterior análisis, el Pareto no dominado se compone de solamente los mejores 100 individuos, lo

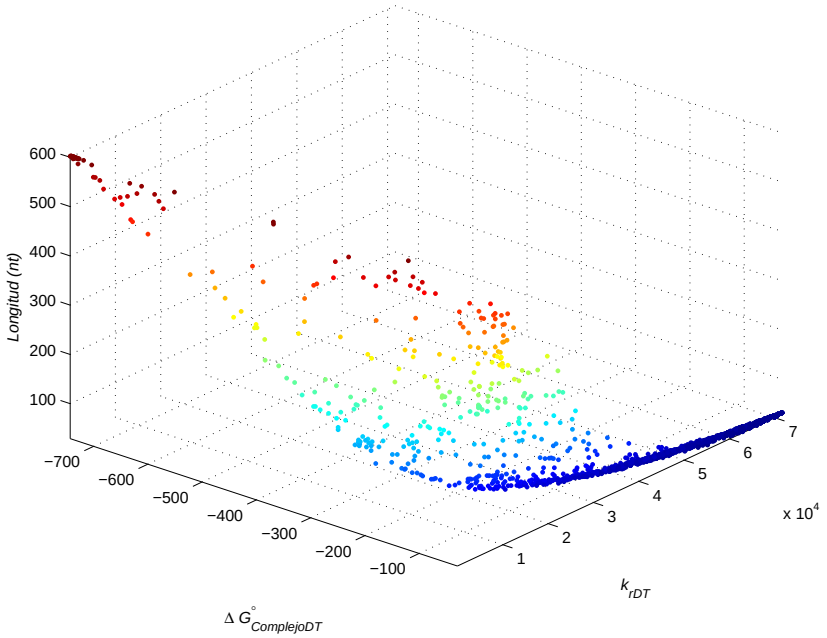



Figura 20: Frente de pareto. Se puede observar el conflicto entre la longitud de los complejos Dx/Tx, la energía libre de los complejos $\Delta G^{\circ}_{ComplejoDT}$ y la constante de disociación del complejo Dx/Tx, $k_{rDT}$.

que asegura que todos los archivos generados por corrida tengan el mismo número de individuos en su última generación.

Una vez ejecutadas las corridas, se obtuvo un frente de Pareto consolidado utilizando las 10 corridas de una misma configuración de parámetros, con la finalidad de obtener un mejor frente, reduciendo así de 1620 a 162 el número total de frentes. Sin embargo, al ser un problema multi-objetivo, la comparación de una configuración con otra no resulta una tarea trivial, principalmente debido a los objetivos en conflicto (Figura 20), por lo que es necesario utilizar métricas que permitan evaluar el desempeño global de cada configuración.

### 5.9.1. Métricas para el desempeño de algoritmos evolutivos multi-objetivo

El diseño de métricas que permitan comparar el desempeño de meta-heurísticas multi-objetivo es una tarea importante que ha sido abordada con anterioridad (Coello-Coello *et al.*, 2010). Según Talbi (2009), los indicadores de desempeño pueden ser clasificados según sus características en:

Indicadores unarios/binarios. Los indicadores binarios permiten comparar directamente dos aproximaciones del verdadero frente de Pareto, mientras que los indicadores unarios asignan a cada aproximación del conjunto óptimo Pareto un valor escalar.

Requerimiento del verdadero frente de Pareto. Algunos indicadores necesitan que el usuario provea el verdadero frente de Pareto del problema, el cual, en la mayoría de los casos, es desconocido.

Necesidad de información extra. Requiere definirse un conjunto de valores que pueden ser difíciles de conseguir, según el caso. (e.g. vector ideal, punto Nadir, solución de referencia, etc.)

En la actualidad existen disponibles una variedad de indicadores de desempeño para diferentes finalidades, por lo que usualmente se utiliza más de un indicador para medir el desempeño de meta-heurísticas multi-objetivo:

Indicadores basados en convergencia. Calculan la proximidad colindante obtenida con respecto al verdadero frente de Pareto. Entre estos se encuentran: el indicador de contribución (Meunier *et al.*, 2000), distancia generacional (Van Veldhuizen, 1999; Van Veldhuizen y Lamont, 1999), indicador-$\epsilon$ (Farhang-Mehr y Azarm, 2002) y el indicador de cubrimiento (Zitzler y Thiele, 1999; Zitzler, 1999).

Indicadores basados en diversidad. Proveen información acerca de la uniformidad de la distribución obtenida en las soluciones a lo largo del frente de Pareto. Algunos ejemplos son: los indicadores de espaciado (Schott, 1995; Van Veldhuizen y Lamont, 1999), esparcimiento (Deb, 2001; Deb *et al.*, 2002) y entropía (Farhang-Mehr y Azarm, 2002).

Indicadores híbridos. Estos intentan medir en un solo valor el desempeño de convergencia y diversidad, como por ejemplo el indicador hipervolumen (Zitzler, 1999) y las métricas R (Hansen, 1998; Knowles y Corne, 2002).

### 5.9.2. Cubrimiento de conjuntos

Debido al desconocimiento del verdadero frente de Pareto del problema, así como sus características específicas, se utiliza el cubrimiento en este trabajo. Este criterio permite comparar un par de conjuntos no dominados, al calcular la fracción por la cual uno de ellos es cubierto por el otro. De esta manera se cumplen tres criterios: distancia, distribución y extensión de datos. Este criterio se define de la siguiente manera: Sean $X', X'' \subseteq X$ dos conjuntos de vectores de decisión. La función cubrimiento $C$ mapea el par ordenado $(X', X'')$ al intervalo $[0, 1]$ (Zitzler y Thiele, 1999; Zitzler, 1999):

$$C(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X' : a' \preceq a''\}|}{|X''|}. \tag{32}$$

El valor $C(X', X'') = 1$ significa que todas las soluciones en $X''$ son dominadas o iguales a las soluciones en $X'$. Por el contrario, $C(X', X'') = 0$ representa la situación en donde ninguna de las soluciones en $X''$ es cubierta por el conjunto $X'$. Nótese que ambas $C(X', X'')$ y $C(X'', X')$ tienen que ser calculadas, debido a que $C(X', X'')$ no es necesariamente igual a $1 - C(X'', X')$.

### 5.9.3. Selección de la mejor configuración de parámetros

Cada uno de los conjuntos de Pareto consolidados obtenidos por configuración fueron comparados todos contra todos utilizando el paquete Guimoo[1], el cual permite calcular métricas de desempeño entre diferentes frentes, incluyendo el criterio de cubrimiento. Sin embargo, este paquete tiene la desventaja de no poder comparar demasiados frentes a la vez, por lo que la comparación se realizó en grupos de 20 conjuntos de Pareto, generando así un total de 9 grupos diferentes, de los que se escogió la mejor configuración de cada uno de ellos. Al final se comparó las mejores configuraciones de cada grupo y se seleccionó aquella con la mejor evaluación (Tabla 5). En la Tabla 6 se describen cada una de las configuraciones ganadoras, donde la mejor de ellas (conf113) es aquella que utiliza cruzamiento uniforme y mutación por complemento como operadores de diversidad, con una probabilidad de 95 % y 1 %, respectivamente, con una población de 1,000 individuos durante 150 generaciones.

Tabla 5: Promedio y desviación estándar de métrica cubrimiento. Las configuraciones mostradas son las mejores de cada grupo. En gris se resalta la configuración con mejor desempeño ($C(A,B)$ con el promedio más alto y $C(B,A)$ con el promedio más bajo). El cálculo de C(A,B) se realizo comparando una configuración (A) contra todas las demas (B). El cálculo de C(B,A) se realiza de igual manera.

| | $C(A,B)$ | | $C(B,A)$ | |
|---|---|---|---|---|
| | $\mu$ | $\pm\sigma$ | $\mu$ | $\pm\sigma$ |
| conf113 | 0.24286075 | 0.031080415 | 0.191586625 | 0.013080743 |
| conf139 | 0.213299 | 0.033788308 | 0.20673075 | 0.025177667 |
| conf13 | 0.23309625 | 0.032963924 | 0.21386725 | 0.033216636 |
| conf142 | 0.223273 | 0.025181765 | 0.2077355 | 0.025545701 |
| conf162 | 0.173560875 | 0.008114124 | 0.252710875 | 0.019971898 |
| conf40 | 0.225228875 | 0.027415082 | 0.230042125 | 0.032647071 |
| conf45 | 0.23285875 | 0.039327042 | 0.2023305 | 0.026668293 |
| conf70 | 0.217848625 | 0.024069155 | 0.229729625 | 0.040139636 |
| conf85 | 0.21041975 | 0.029878712 | 0.237712625 | 0.038313567 |

[1]http://guimoo.gforge.inria.fr/

Tabla 6: Descripción de configuraciones ganadoras. La información se muestra según el operador de cruzamiento (Op. Cruz) y de mutación (Op. Mut.), probabilidades de cruzamiento (Prob. Cruz) y mutación (Prob. Mut.), población de padres e hijos (N.P. y N.H.) y el número de generaciones (N. Gen.).

| | Op. Cruz. | Op. Mut. | Prob. Cruz. | Prob. Mut. | N. P. | N. H. | N. Gen. |
|---|---|---|---|---|---|---|---|
| conf13 | Un punto | Inversión | 0.95 | 0.05 | 1000 | 1000 | 250 |
| conf40 | Un punto | Complemento | 0.95 | 0.05 | 1000 | 1000 | 250 |
| conf45 | Un punto | Complemento | 0.85 | 0.1 | 1000 | 1000 | 250 |
| conf70 | Un punto | Deleción | 0.9 | 0.1 | 1000 | 1000 | 250 |
| conf85 | Uniforme | Inversión | 0.95 | 0.05 | 500 | 500 | 500 |
| conf113 | Uniforme | Complemento | 0.95 | 0.1 | 500 | 500 | 500 |
| conf139 | Uniforme | Deleción | 0.95 | 0.05 | 500 | 500 | 500 |
| conf142 | Uniforme | Deleción | 0.9 | 0.1 | 500 | 500 | 500 |
| conf162 | Uniforme | Deleción | 0.85 | 0.1 | 2000 | 2000 | 150 |

### 5.9.4. Selección de sondas

El siguiente paso es seleccionar del frente ganador las sondas que se usarán en la simulación. El objetivo es medir el desempeño de cada complejo diagnóstico/tratamiento, por lo que es necesario seleccionar un individuo representativo por cada tipo de mutación (extremos con nucleótidos sobresalientes y bucles interiores). Para hacer la competencia equitativa, las sondas seleccionadas tienen la misma longitud en nucleótidos. Esta longitud se estableció en 35 pares de bases debido a que fue la longitud mínima donde se encontraron 3 individuos con estructuras diferentes. La Tabla 7 muestra los complejos seleccionados.

Para visualizar las estructuras de los complejos, se utilizó el paquete Nupack (Zadeh *et al.* (2011)), el cual permite el análisis y diseño de sistemas de ácidos nucleicos. Entre todas sus funciones, este software genera representaciones gráficas de complejos, partiendo de un archivo de texto. Las representaciones gráficas generadas se pueden apreciar en la Figura 21.

Tabla 7: Individuos seleccionados para simulación estocástica.

| Nombre del complejo | Marcador de mutación | Configuración |
|---|---|---|
| Tipo I | Extremo sobresaliente 5′ (señal Dx) | [120, 154, 0, :, 3, :, [, 32, ]] |
| Tipo II | Bucle interior | [94, 128, 26, [, 26, ], (, 3, ), [, 6, ]] |
| Tipo III | Extremo sobresaliente 3′ (señal Dx) | [88, 122, 32, :, 1, :, [, 31, ], ;, 3, ;] |

A) Complejo tipo I. B) Complejo tipo II. C) Complejo tipo III.

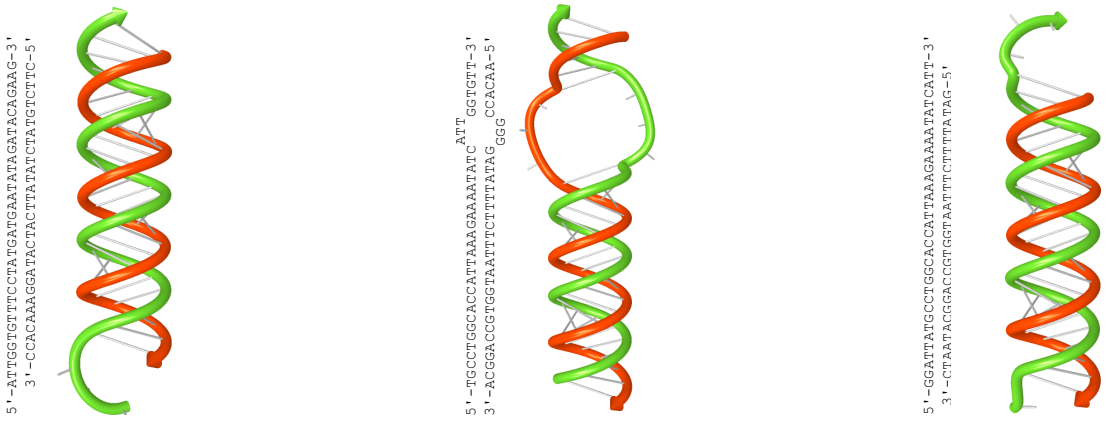



Figura 21: Secuencia y representación gráfica de complejos Dx/Tx. Cada complejo se compone de una señal de diagnóstico (verde) y una señal de tratamiento (rojo). Los nucleótidos desapareados localizados en la sonda Dx contienen el marcador de la mutación $\Delta F508$. La representación gráfica fue generada por el software Nupack (Zadeh *et al.*, 2011).

# Capítulo 6. Simulación estocástica y desempeño de sondas

Una vez terminado el diseño de los complejos, la siguiente tarea a realizar sería corroborar su desempeño en laboratorio. Sin embargo, debido a la falta de infrestructura para realizar estos experimentos, se propone verificar el desempeño de los mismos en un ambiente simulado por computadora utilizando métodos Monte Carlo. Para esto, se presentan dos modelos de simulación, un modelo termodinámico de detección y otro para detección y tratamiento, utilizando los parámetros y condiciones que asemejen lo más posible un experimento real. Cada modelo es simulado con el algoritmo de Gillespie (Gillespie, 1977) utilizando el paquete de software Dizzy (Ramsey *et al.*, 2005) el cual implementa este algoritmo.

## 6.1. Modelo termodinámico de detección

El objetivo de este modelo es predecir la capacidad de detección de las sondas ante la presencia de moléculas de ARNm que contengan la deleción ΔF508 (Figura 22). Para esto es necesario especificar todas las moléculas y reacciones necesarias para que se lleve a cabo dicha detección. Una molécula de ARN mensajero mutado se representa como ARNm mutado, un complejo molecular de diagnóstico/tratamiento se representa como Complejo $_{\mathrm{Dx/Tx}}$, el cual a su vez se conforma por dos sondas de ADN, una que sirve como señal de diagnóstico (representado como $\mathrm{Señal_{Dx}}$) y la otra funciona como señal de tratamiento ($\mathrm{Señal_{Tx}}$). Esta reacción se puede definir de la siguiente manera:

$$\text{Complejo }_{\mathrm{Dx/Tx}} \underset{k_{caso}}{\overset{k_{cdiso}}{\rightleftharpoons}} \mathrm{Señal_{Dx}} + \mathrm{Señal_{Tx}}, \tag{33}$$

donde $k_{cdiso}$ y $k_{caso}$ son las constantes de cinética molécular de disociación y asociación, respectivamente. Cuando un complejo diagnóstico/tratamiento colisiona con una molécula de ARNm mutado, el mayor número de nucleótidos complementarios entre la señal de diagnóstico Dx y el ARN mutado hace que este nuevo complejo ADN/ARN sea termodinámicamente más favorable, liberando como consecuencia la señal de tratamiento Tx. Este desplazamiento de cadenas es mediado por las secciones no complementarias del complejo Dx/Tx, por lo que el desplazamiento se puede modelar en dos fases: en primer lugar, las secciones no complementarias del complejo Dx/Tx se asocian a su respectiva secuencia complementaria en el ARNm mutado generando un complejo intermedio llamado unión$_{\text{ADN/ARNm}}$,

$$\text{ARNm}_{\text{mutado}} + \text{Complejo}_{\text{ Dx/Tx}} \underset{k_{unbind}}{\overset{k_{bind}}{\rightleftharpoons}} \text{unión}_{\text{ADN/ARNm}}, \tag{34}$$

donde $k_{bind}$ y $k_{unbind}$ son las constantes de cinética de unión y separación de complejos. Posteriormente, los nucleótidos complementarios de las señales Dx y Tx se disocian del complejo, al tiempo que Dx se asocia con la molécula de ARNm mutado, de manera análoga a una bragueta, por lo que los complejos unión$_{\text{ADN/ARNm}}$ terminan por disociarse en complejos ADN/ARN y moléculas Tx,

$$\text{unión}_{\text{ADN/ARNm}} \xrightarrow{k_{des}} \text{ADN/ARN} + \text{Tx}, \tag{35}$$

donde $k_{des}$ es la constante de cinética de desplazamiento de cadenas, y el producto final esperado es un número de moléculas Tx igual a la cantidad de moléculas iniciales de complejos Dx/Tx.

Por otro lado, cuando complejos ADN/ARN colisionan con moléculas Tx, ocurre

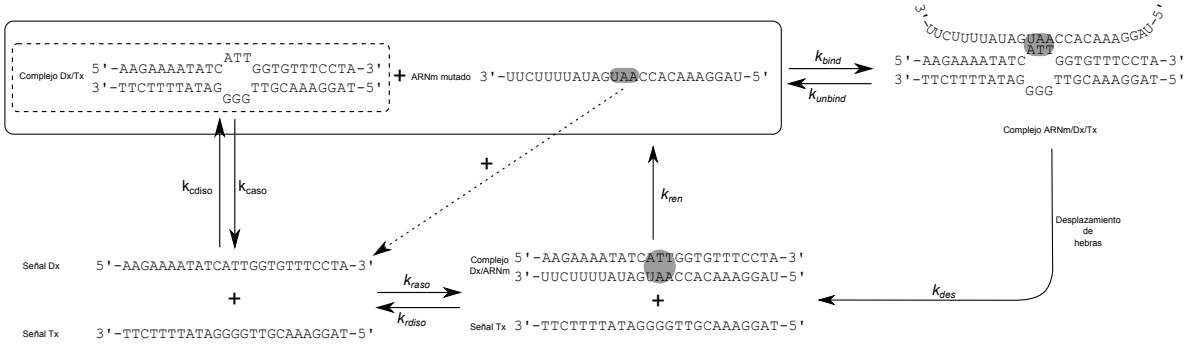



Figura 22: Esquema del modelo termodinámico de detección positivo. Los complejos Dx/Tx puede disociarse en señales Dx y Tx, a su vez, cuando estas señales colisionan, la complementariedad de bases provoca que estas hibríden formándose complejos Dx/Tx. Ante la presencia de ARNm mutado, se inicia la primera fase del desplazamiento de cadenas dirigido por la secciones complementarias entre el complejo Dx/Tx y las moléculas de ARNm mutado (sombreado en gris), generándose un complejo intermedio. Por ultimo, la diferencia de energías generada entre complejos provoca un desplazamiento de cadenas, liberándose la señal de diagnóstico.

una reacción inversa a la anterior, liberándose complejos Dx/Tx y moléculas ARNm mutado,

$$\text{ADN/ARN} + \text{Tx} \xrightarrow{k_{ren}} \text{Complejo}_{\text{Dx/Tx}} + \text{ARNm}_{\text{mutado}}, \tag{36}$$

donde $k_{ren}$, la constante de cinética de restitución, se espera que sea mucho menor que $k_{des}$. A su vez, un complejo ADN/ARN se puede disociar en moléculas de ARNm mutado y señales Dx,

$$\text{ADN/ARN} \underset{k_{raso}}{\overset{k_{rdiso}}{\rightleftharpoons}} \text{ARNm}_{\text{mutado}} + \text{Dx}, \tag{37}$$

donde $k_{rdiso}$ y $k_{raso}$ son las constantes de cinética de disociación y asociación de complejos ADN/ARN.

En ausencia de moléculas de ARNm mutadas, no debería ocurrir desplazamiento alguno por medio de complejos Dx/Tx, ya que el marcador de la mutación en la señal Dx no sería complementaria con alguna otra molécula. Sin embargo, al ser una reac-

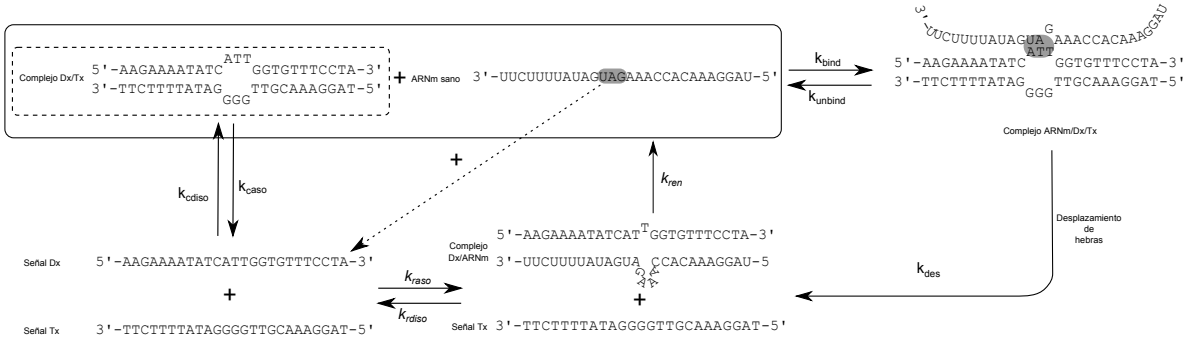



Figura 23: Esquema de modelo termodinámico de detección negativo. Los complejos Dx/Tx puede disociarse en señales Dx y Tx, a su vez, cuando estas señales colisionan, la complementariedad de bases provoca que estas hibríden formándose complejos Dx/Tx. Ante la presencia de ARNm sano, las bases que sean complementarias en la sección no complementaria del complejo Dx/Tx (sombreado en gris) iniciarán la primera fase del desplazamiento, generándose un complejo intermedio poco estable. En caso que la diferencia de energías entre complejos sea favorable, se produce un desplazamiento de cadenas, liberándose la señal de tratamiento y produciéndose un complejo ADN/ARN semicomplementario.

ción reversible, los complejos Dx/Tx pueden disociarse y generar un número de falsos positivos, situación que se desa minimizar.

De manera análoga, intercambiando las moléculas de ARNm mutado por moléculas de ARNm sano, se obtiene el modelo de detección negativo:

$$
\begin{aligned}
\text{Complejo}_{\text{Dx/Tx}} &\underset{k_{caso}}{\overset{k_{cdiso}}{\rightleftharpoons}} \text{Dx} + \text{Tx} \\
\text{ARNm}_{\text{sano}} + \text{Complejo}_{\text{Dx/Tx}} &\underset{k_{unbind}}{\overset{k_{bind}}{\rightleftharpoons}} \text{unión}^{fp}_{\text{ADN/ARN}} \\
\text{unión}^{fp}_{\text{ADN/ARN}} &\xrightarrow{k_{des}} \text{ADN/ARN}_{\text{fp}} + \text{Tx} \\
\text{ADN/ARN}_{fp} + \text{Tx} &\xrightarrow{k_{ren}} \text{Complejo}_{\text{Dx/Tx}} + \text{ARNm}_{\text{sano}} \\
\text{ADN/ARN}_{fp} &\underset{k_{raso}}{\overset{k_{rdiso}}{\rightleftharpoons}} \text{ARNm}_{\text{sano}} + \text{Dx},
\end{aligned}
\tag{38}
$$

donde $\text{ADN/ARN}_{fp}$ es el complejo ADN/ARN generado por detección errónea (falso positivo) de una sonda específica. Suponiendo una configuración óptima, se esperaría que el número de falsos positivos fuera mínimo y el número de moléculas de ARNm sano se mantuviera constante respecto el tiempo.

## 6.2. Simulación del modelo termodinámico de detección

Basado en el modelo termodinámico de detección, se generaron seis escenarios de simulación, tres con el modelo positivo y tres con el modelo negativo por cada configuración de complejo seleccionado en el capítulo anterior (Tabla 7). El objetivo del modelo positivo es simular la eficiencia de detección y generación de tratamiento de los complejos Dx/Tx en presencia de moléculas de ARNm que contengan la deleción $\Delta F508$ de la fibrosis quística. En cambio, la finalidad del modelo negativo es simular el comportamiento de los complejos Dx/Tx en ausencia de la mutación en moléculas de ARNm sanas, por lo que se esperaría que no se genere un tratamiento.

Cada modelo contiene las mismas reacciones, especímenes y constantes cinéticas anteriormente descritas, exceptuando el número de partículas tanto en complejos Dx/Tx como en ARNm celular, los cuales son los parámetros a establecer. Para este trabajo, se decidió fijar estos parámetros en 100, 1,000, 10,000 y 100,000 moléculas tanto para los complejos Dx/Tx como el ARNm. Además, por tratarse de un algoritmo Montecarlo, cada simulación fue ejecutada 10 veces de manera independiente. El tiempo de simulación se estableció en 18 horas, realizándose un muestreo cada segundo. Los resultados por configuración se presentan a continuación.

### 6.2.1. Estabilidad de complejos diagnóstico/tratamiento

El objetivo de esta simulación es predecir la estabilidad de los complejos Dx/Tx. Con estabilidad se entiende a la capacidad de los complejos Dx/Tx de mantenerse hibridados con respecto al tiempo bajo condiciones fisiológicas ideales (Ph 7.0 y temperaturas de incubación de 37°C). Los resultados obtenidos se presentan en número de moléculas diagnóstico y tratamiento presentes (complejos separados) contra el tiempo

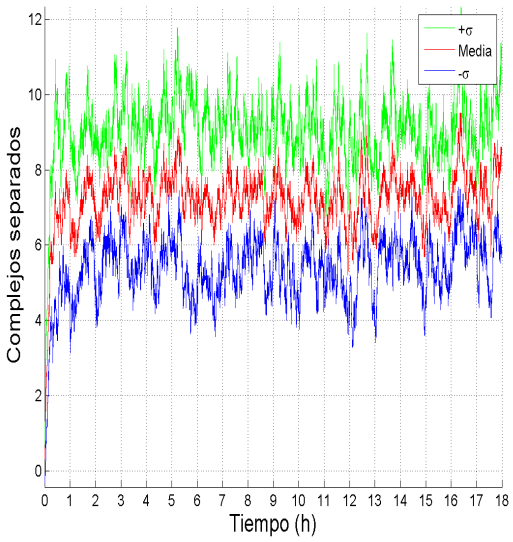
de simulación utilizando el promedio de las 10 corridas diferentes (Figuras 24 a 26).

Debido a que la longitud de los complejos es la misma (35 nt), y la región apareada contiene la misma cantidad de nucleótidos (con excepción del complejo tipo III), los resultados arrojaron un comportamiento parecido, por lo que el análisis se realiza pensando en los diferentes complejos en conjunto. En el primer escenario (100 moléculas), las simulaciones presentaron un promedio de 7.3 complejos separados con una desviación estándar de 0.78, lo que se traduce a un error del 0.073 %. En simulaciones con 1,000 complejos, lograron separarse un promedio de 24.5 complejos, generándose un error del 2.45 % ($\pm$0.143 %). Por otro lado, las simulaciones con 10,000 moléculas arrojaron un promedio 78.71 señales Tx, lo que significa un error del 0.7871 % ($\pm$0.0253 %). Por último, las simulaciones realizadas con 100,000 moléculas cuentan con un promedio de 250.05 complejos separados, esto es, un error del 0.25 % ($\pm$0.00497 %). Comparando estos resultados con los presentados en Martínez-Pérez (2007), donde los complejos Dx/Tx alcanzaron una separación espontánea del 13.1 % en una concentración de $1\mu M$, se puede observar el potencial del optimizador para el diseño de complejos que permanezcan estables bajo condiciones fisiológicas.

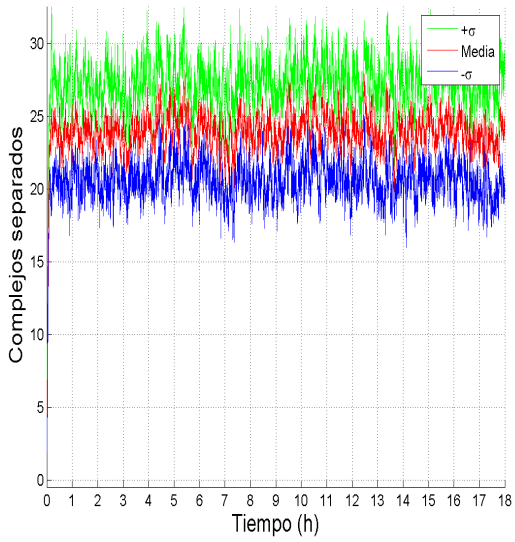
### 6.2.2. Complejo tipo I

Para estudiar el comportamiento hipotético del complejo tipo I, se simularon tres escenarios dados por el número de complejos Dx/Tx establecidos anteriormente. En el primer escenario, tanto el modelo positivo (Figura 27A) como el modelo negativo (Figura 27B) se simularon con el mismo número de moléculas de complejos Dx/Tx y ARNm celular (1,000 moléculas). Como se observa, el número de complejos que realizan una detección correcta (verdaderos positivos) es de $409 \pm 2.5$ moléculas aproximadamente, en contraste con el número de falsos positivos, el cual es aproximadamanete

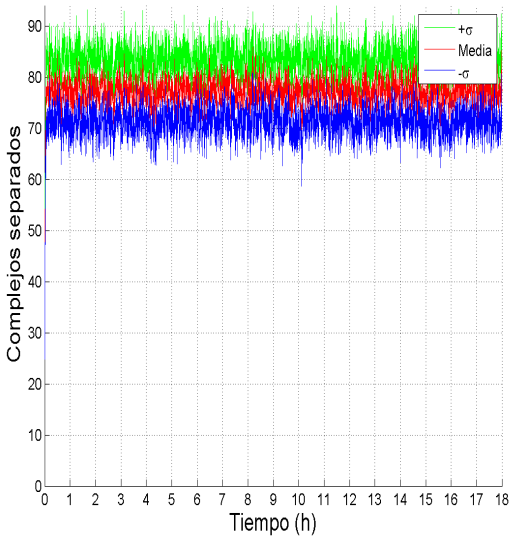
A) 100 moléculas




B) 1,000 moléculas




C) 10,000 moléculas




D) 100,000 moléculas

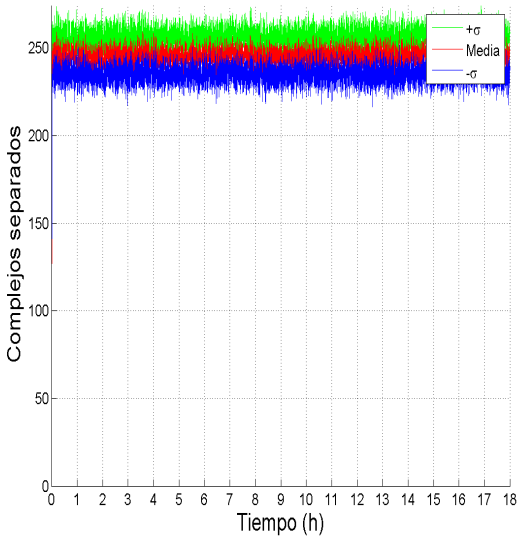
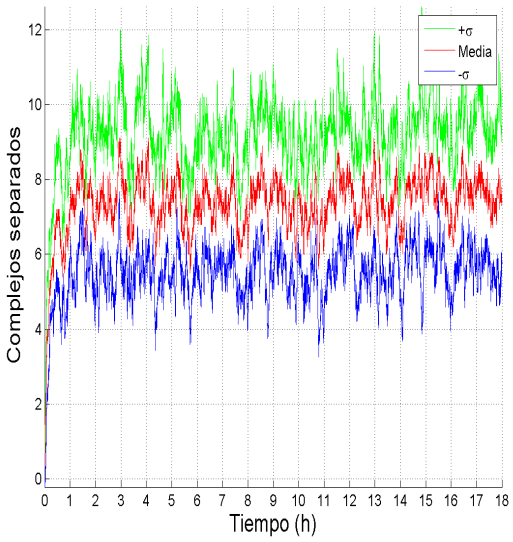
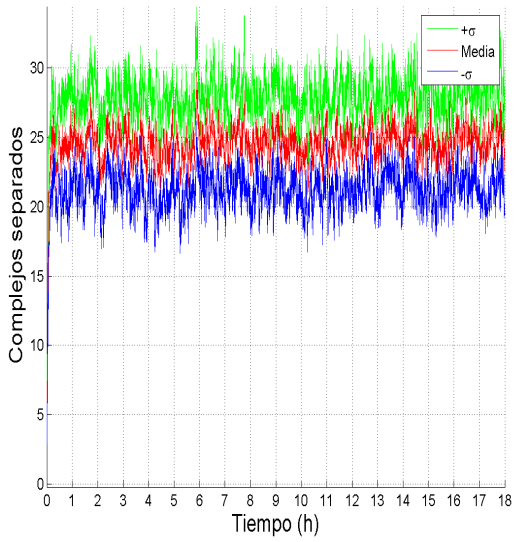



Figura 24: Resultados de simulación: estabilidad de complejos diagnóstico/tratamiento (complejo tipo I).

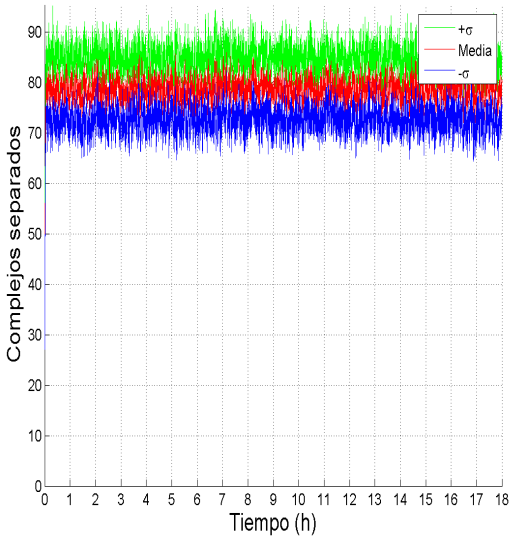
A) 100 moléculas




B) 1,000 moléculas




C) 10,000 moléculas




D) 100,000 moléculas

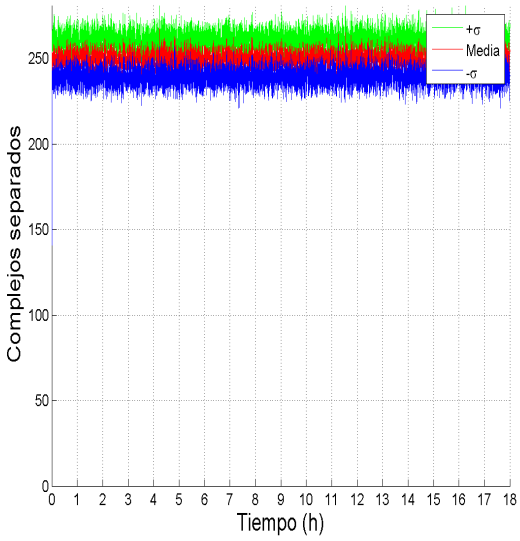
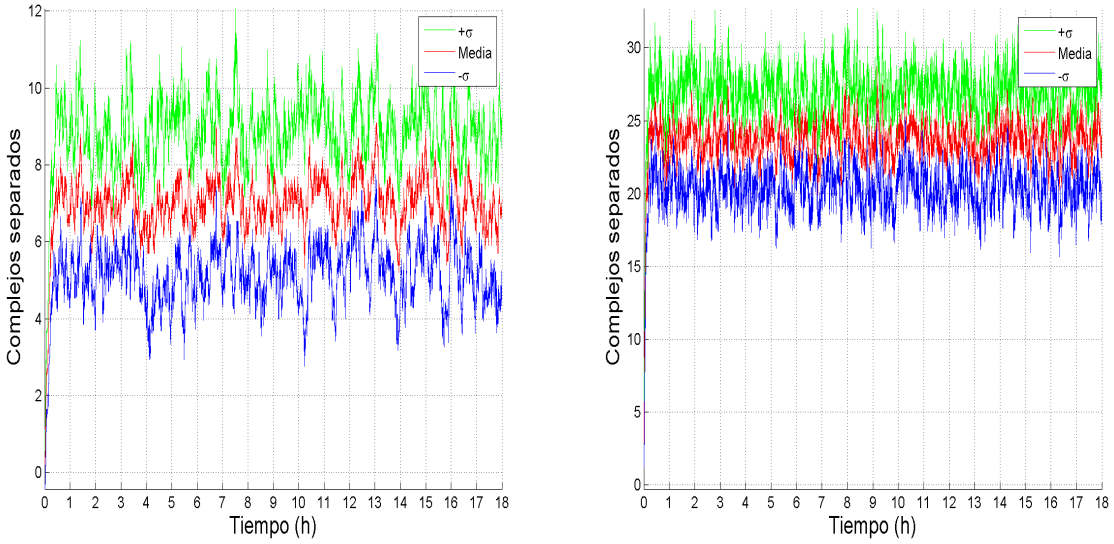



Figura 25: Resultados de simulación: estabilidad de complejos complejos diagnóstico/tratamiento. (complejo tipo II).

A) 100 moléculas

B) 1,000 moléculas




C) 10,000 moléculas

D) 100,000 moléculas

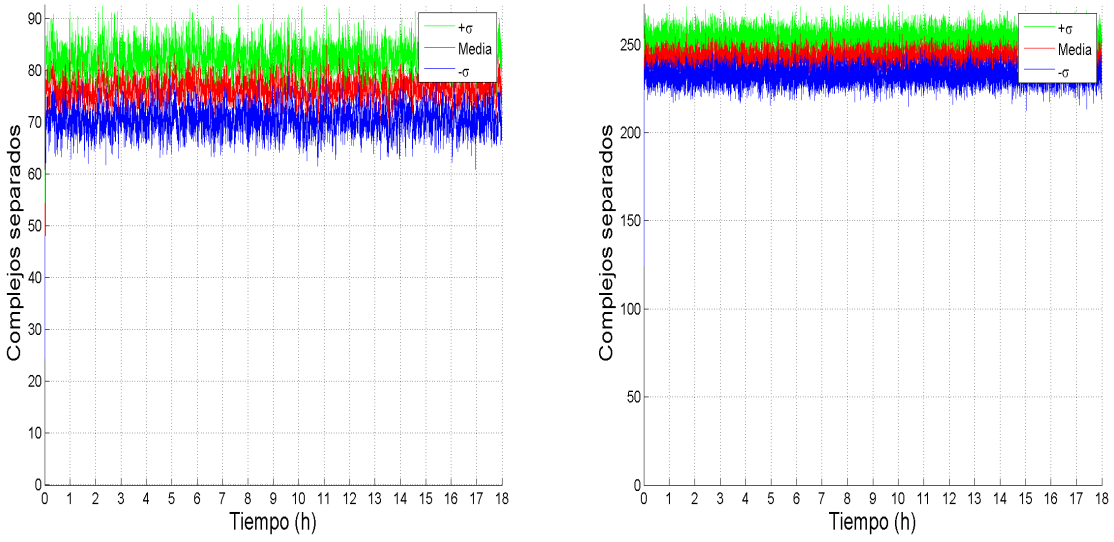



Figura 26: Resultados de simulación: estabilidad de complejos diagnóstico/tratamiento. (complejo tipo III).

$74 \pm 1.23$ moléculas. Esto quiere decir que, de un total de 1,000 moléculas, se obtiene una sensibilidad y una especificidad del 40.9 % y 92.6 %, respectivamente. En el segundo escenario, los resultados obtenidos muestran un promedio de $4,095 \pm 8$ verdaderos positivos de un total de 10,000 moléculas (Figura 27C)) contra $310 \pm 2.6$ falsos positivos (Figura 27D), otorgando una sensibilidad del 40.95 % contra una especificidad del 96.9 % .

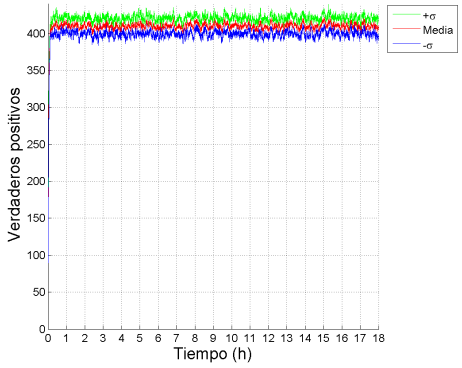
Por último, en el tercer escenario se contabilizan $40,966 \pm 25.6$ verdaderos positivos (Figura 27E) contra $1150 \pm 5.3$ falsos positivos (Figura 27F), de un total de 100,000 moléculas, generando una sensibilidad y especificidad del 40.96 % y 98.85 %, respectivamente. Como se puede observar, a mayor número de moléculas Dx/Tx, mayor la especificidad de las sondas. Sin embargo, la sensibilidad se mantiene constante, esto debido a que en todos los experimentos se utilizó la misma cantidad de moléculas tanto en complejos Dx/Tx como en ARNm. Si se deseara aumentar la sensibilidad, bastaría con aumentar la concentración de complejos respecto a la de moléculas de ARNm. No obstante, con esto se comprometería la especificidad, ya que al haber mayor número de moléculas Dx/Tx, la probabilidad de que hibriden moléculas de manera errónea aumenta, disminuyendo como consecuencia la especificidad, por lo que es necesario encontrar un compromiso entre estas dos variables.

### 6.2.3. Complejo tipo II

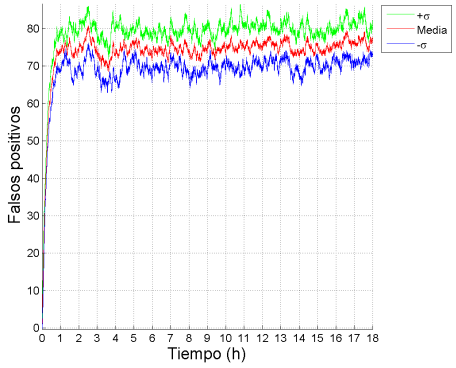
Para poder evaluar de manera justa el comportamiento de este complejo, las simulaciones se realizaron tomando en cuenta los mismos tres escenarios del caso anterior. En el primer escenario, se realizaron 10 simulaciones con 1,000 moléculas de complejos Dx/Tx y ARNm para el modelo positivo (Figura 28A) y el modelo negativo (Figura 28B). Los resultados obtenidos arrojaron un promedio de $413 \pm 2.6$ verdaderos

1,000 moléculas.

A) Modelo positivo

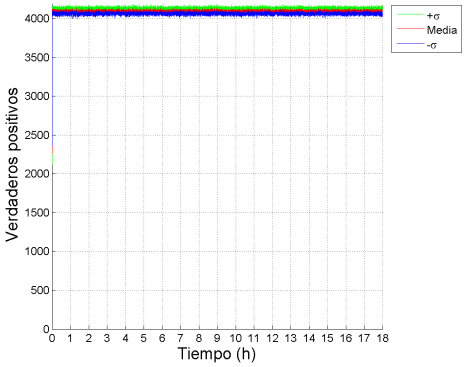



B) Modelo negativo

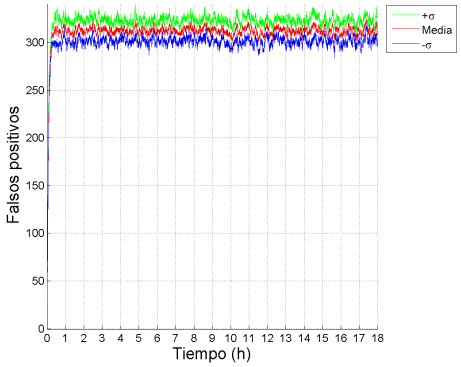



10,000 moléculas.

C) Modelo positivo

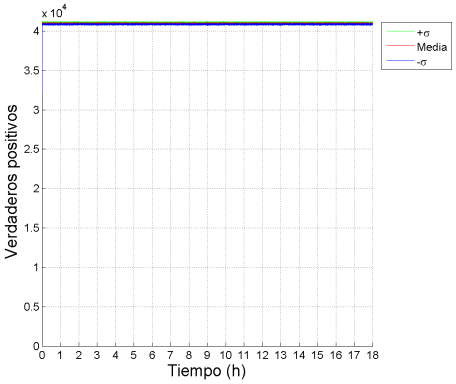



D) Modelo negativo




100,000 moléculas.

E) Modelo positivo




F) Modelo negativo

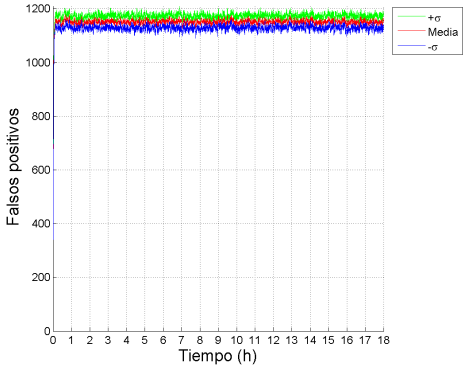



Figura 27: Resultados de simulación: modelo termodinámico de detección (complejo tipo I).

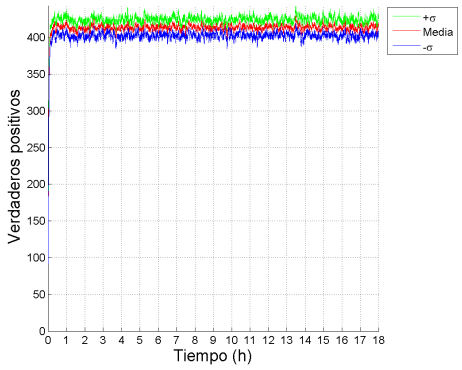
positivos contra $74 \pm 1.1$ falsos positivos, respectivamente, lo cual se traduce en una sensibilidad del 41.3 % y una especificidad del 92.6 %. En el segundo escenario se utilizaron 10,000 moléculas de sondas y ARNm, produciéndose $4,137 \pm 8$ verdaderos positivos (Figura 28C) contra $320 \pm 2.6$ falsos positivos (Figura 28D), por lo que se logra una sensibilidad del 41.37 % y una especificidad del 96.8 %. Con respecto al tercer escenario, se cuenta con un promedio de $41,378 \pm 25.5$ verdaderos positivos (Figura 28E) contra $1,189 \pm 5.5$ falsos positivos (Figura 28F), de un total de 100,000 moléculas, lo que se traduce en un 41.37 % en especificidad contra un 98.8 % en sensibilidad. A simple vista se puede observar el mismo comportamiento que la configuración anterior, esto es, a mayor número de moléculas, mejor la especificidad junto con un estancamiento en la sensibilidad (alrededor del 40 %).

### 6.2.4. Complejo tipo III

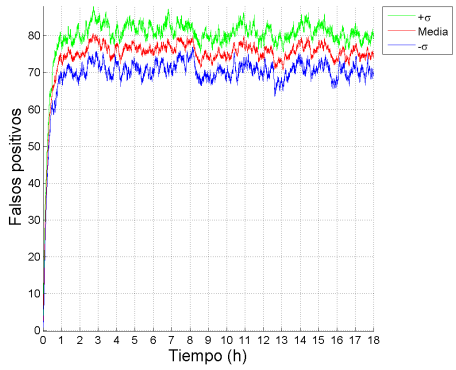
Los resultados de simulación para este complejo se presentan en tres escenarios (Figura 29). En el primer escenario, se ejecutaron ambos modelos positivo y negativo con 1,000 moléculas de complejos Dx/Tx y ARNm, resultando en un promedio de $425 \pm 2.6$ verdaderos positivos (Figura 29A) contra $408 \pm 2.6$ falsos positivos (Figura 29B), por lo que se tiene una sensibilidad y una especificidad del 42.5 % y 59.2 %, respectivamente. En el segundo escenario, se obtuvieron un promedio de $4,262 \pm 8.15$ verdaderos positivos (Figura 29C) contra $4,090 \pm 8.12$ falsos positivos (Figura 29D), de un total de 10,000 moléculas, alcanzando una especificidad del 59.1 % y una sensibilidad del 42.62 %. Por último, en el escenario con 100,000 moléculas, se generaron $42,633 \pm 25.7$ verdaderos positivos (Figura 29E) contra $40,911 \pm 25.46$ falsos positivos (Figura 29F), lo cual genera una especificidad del 42.63 % contra una sensibilidad del 59.09 %.

1,000 moléculas.

A) Modelo positivo

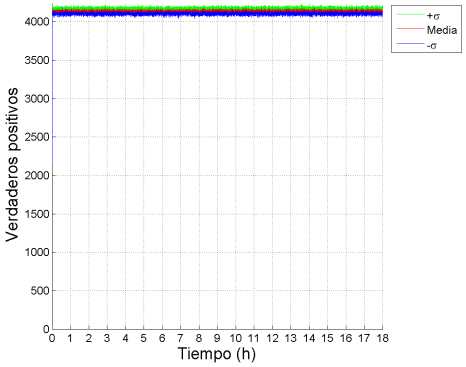



B) Modelo negativo

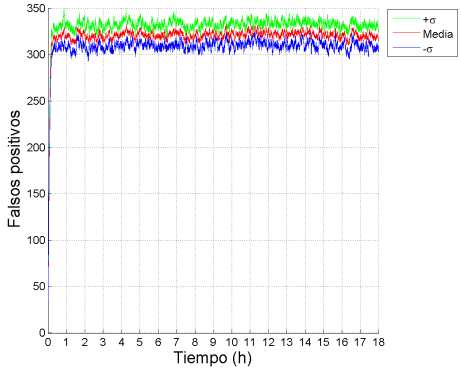



10,000 moléculas.

C) Modelo positivo

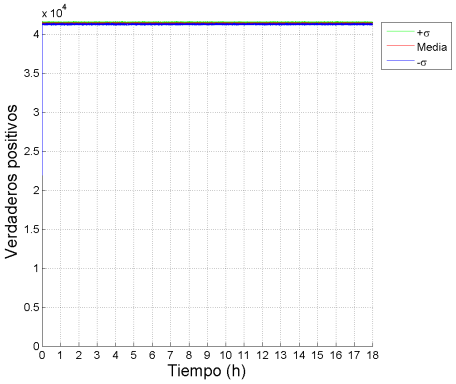



D) Modelo negativo




100,000 moléculas.

E) Modelo positivo




F) Modelo negativo

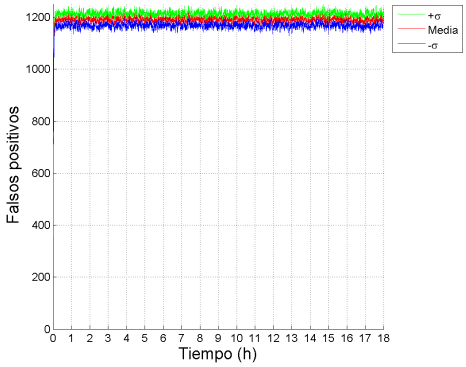
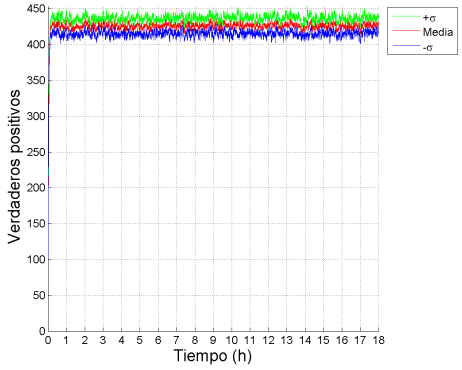


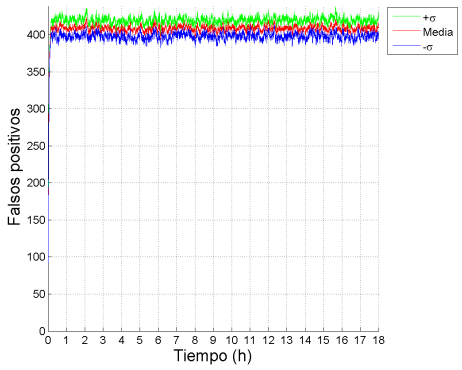

Figura 28: Resultados de simulación: modelo termodinámico de detección (complejo tipo II).

1,000 moléculas.

A) Modelo positivo

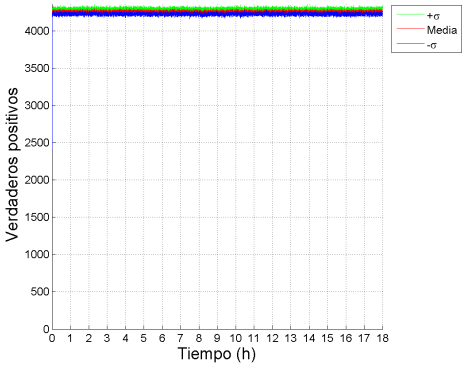



B) Modelo negativo

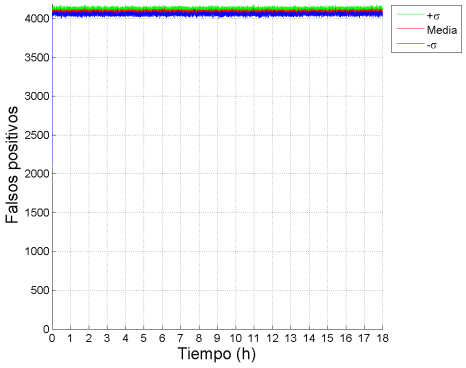



10,000 moléculas.

C) Modelo positivo

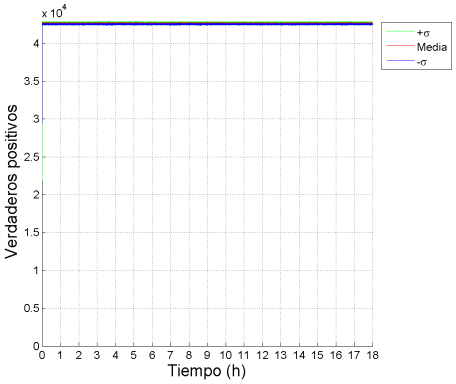



D) Modelo negativo




100,000 moléculas.

E) Modelo positivo




F) Modelo negativo

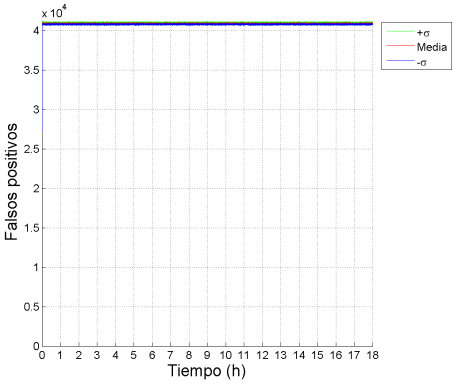



Figura 29: Resultados de simulación: modelo termodinámico de detección (complejo tipo III).

Tabla 8: Especificidad y sensibilidad obtenidos por complejo.

| Configuración | Moléculas | Especificidad | Sensibilidad |
|---|---|---|---|
| Tipo I | 1,000 | 92.6 % | 40.9 % |
| | 10,000 | 96.9 % | 40.95 % |
| | 100,000 | 98.85 % | 40.96 % |
| Tipo II | 1,000 | 92.6 % | 41.3 % |
| | 10,000 | 96.8 % | 41.37 % |
| | 100,000 | 98.8 % | 41.37 % |
| Tipo III | 1,000 | 42.5 % | 59.2 % |
| | 10,000 | 59.1 % | 42.62 % |
| | 100,000 | 59.09 % | 42.63 % |

### 6.2.5. Análisis de resultados

En la Tabla 8 se presenta un resumen comparativo de los complejos Dx/Tx en términos de sensibilidad y especificidad. Se puede observar que los complejos tipo I y II tuvieron un desempeño similar en ambos rubros para cada uno de los escenarios simulados. Sin embargo, revisando minuciosamente los resultados, se observa que la configuración tipo II tuvo una mejor sensibilidad en todos los escenarios con respecto a la configuración tipo I ($\approx$ 0.04 %). El desempeño en especificidad es mucho más parejo, donde la configuración tipo I fue ligeramente mejor en los escenarios con 10,000 y 100,000 moléculas, pero no así para el escenario con 1,000 moléculas, donde comparte el mismo desempeño que la configuración tipo II. En cualquier caso, la diferencia no es más del 0.1 %.

Por otro lado, analizando los resultados del complejo tipo III se puede deducir que este complejo obtuvo el peor desempeño en las simulaciones, y aunque tuvo ligeramente una mayor sensibilidad que el complejo tipo II en todos los escenarios (alrededor de 18.3 % en el primer escenario y 1.67 % en los restantes), la especificidad del complejo se encuentra claramente comprometida al obtener una diferencia de 39 puntos porcentuales menos con respecto a los otros complejos en cada escenario. Al revisar la

estructura de este complejo (Figura 21C), encontramos que a diferencia de las otras configuraciones, la señal Dx cuenta con un nucleótido sobresaliente en su extremo 5′, así como con 3 nucleótidos sobresalientes por su extremo 3' (donde se lleva a cabo la detección). A nivel de parámetros, esta configuración cumple con todas las restricciones de optimización, esto es, estabilidad a nivel complejo Diagnóstico/Tratamiento (Dx/Tx), estabilidad en el complejo Dx/ARNm mutado y una estabilidad menor en complejo Dx/ARNm sano (Tabla 9). Sin embargo, a nivel de simulación, el comportamiento demuestra lo contrario, por lo que el problema al parecer radica en esa sección adicional no complementaria de la sonda. De acuerdo al modelo de termodinámico de detección (Figura 22), para que un desplazamiento de hebras pueda llevarse a cabo, es necesario que primero hibriden las regiones complementarias localizadas en el vecindario donde ocurre la mutación del ARNm. Note que, en el modelo negativo de este complejo, la hibridación no puede ocurrir por el extremo 3' de la sonda Dx, ya que no existe un vecindario complementario en una molécula de ARNm sana, pero si puede ocurrir una hibridación rápida por el extremo 5′. Esto se debe a que en el modelo termodinámico del vecino más cercano (SantaLucia y Hicks, 2004), se genera un evento de hibridación a partir de dos nucleótidos, otorgando energía suficiente al complejo Dx/ARNm sano para que se lleve a cabo un desplazamiento de hebras. Una solución a este problema consiste en variar la concentración de los especímenes, ya sea aumentando o disminuyendo el número de moléculas de sondas o ARNm, disminuyendo así la probabilidad de que un complejo Dx/Tx hibride con una molécula de ARNm sano; y en el caso de que esto suceda, aumentar la probabilidad de que el complejo ADN/$ARN_{fp}$ hibride con una molécula Tx, regresando el complejo a su estado original.

Tabla 9: Energias libres generadas por tipo de complejo.

| Nombre del complejo | $\Delta G^\circ$ complejo Dx/Tx | $\Delta G^\circ$ complejo ADN/ARNm mutado | $\Delta G^\circ$ complejo ADN/ARNm sano |
|---|---|---|---|
| Tipo I | -36.75 | -36.6 | -32.1 |
| Tipo II | -34.16 | -39.1 | -32.3 |
| Tipo III | -38.22 | -36.1 | -32.3 |

## 6.3. Modelo termodinámico de diagnóstico y terapia

Hasta el momento se ha definido el modelo encargado de la detección de moléculas de ARNm que contenga una deleción, el cual en teoría el cual podría ser utilizado tanto en ambientes de laboratorio (*in vitro*) como en ambientes celulares (*in vivo*). Sin embargo, si se desea desarrollar un modelo completo de un gen computacional que permita tanto el diagnóstico como el tratamiento de la fibrosis quística, hay ciertos factores adicionales que considerar, principalmente relacionados con ambientes *in vivo*, como lo son los procesos de transcripción y traducción en la célula, esenciales también para un diseño óptimo del autómata molecular.

### 6.3.1. Detección de ARN mutado y liberación de la señal de tratamiento

De acuerdo al modelo descrito en la sección anterior, el producto final del proceso de detección positivo son las señales de tratamiento Tx. Esta molécula será la encargada de activar el auto-ensamblamiento del gen computacional.

### 6.3.2. Auto-ensamblamiento del gen computacional

Para modelar este subproceso, se requieren cuatro especímenes de moléculas: las dos partes del gen computacional desactivado (definidos como Gen computacional$_A$ y Gen computacional$_B$, respectivamente), la señal de tratamiento Tx y, como producto final, el gen computacional completo unido mediante la señal de tratamiento Tx (defi-

nido como Gen computacional$_U$). La siguiente cadena de reacciones describe el proceso completo

$$\begin{aligned}
\text{Gen computacional}_\text{A} + \text{Tx} &\underset{k_{des_p1}}{\overset{k_{ens_p1}}{\rightleftharpoons}} \text{Gen computacional}_\text{A/TX} \\
\text{Gen computacional}_\text{B} + \text{Tx} &\underset{k_{des_p2}}{\overset{k_{ens_p2}}{\rightleftharpoons}} \text{Gen computacional}_\text{B/Tx} \\
\text{Gen computacional}_\text{A/TX} + \text{Gen computacional}_\text{B} &\underset{k_{des_p3}}{\overset{k_{ens_p3}}{\rightleftharpoons}} \text{Gen computacional}_\text{U} \\
\text{Gen computacional}_\text{B/TX} + \text{Gen computacional}_\text{A} &\underset{k_{des_p4}}{\overset{k_{ens_p4}}{\rightleftharpoons}} \text{Gen computacional}_\text{U},
\end{aligned} \tag{39}$$

donde $k_{ens\_p1}$, $k_{ens\_p2}$, $k_{ens\_p3}$ y $k_{ens\_p4}$ son las constantes de cinética molecular de asociación, y $k_{des\_p1}$, $k_{des\_p2}$, $k_{des\_p3}$, $k_{des\_p4}$ las constantes de cinética molecular de disociación respectivas.

Observe que el estado más estable de la cadena de reacción es el complejo Gen computacional$_\text{U}$, debido principalmente al mayor número de nucleótidos apareados (Figura 30B)). Sin embargo, para que el gen computacional pueda completarse, necesita la ayuda de la enzima ligasa, encargada de unir covalentemente cada una de las partes del gen artificial. Esta reacción podría modelarse con el esquema clásico de la cinética enzimática, en el que una enzima se une al sustrato para formar un complejo enzima/sustrato como requerimiento para la etapa catalítica. Para simplificar el proceso, el ligamiento del gen computacional se modeló como:

$$\text{Gen computacional}_\text{U} \xrightarrow{k_{lig}} \text{Gen computacional}_\text{inactivo}, \tag{40}$$

donde Gen computacional$_\text{U}$ es el gen computacional no ligado, Gen computacional$_\text{inactivo}$ es el gen computacional ligado en su estado inactivo y $k_{lig}$ es la constante de velocidad de reacción de la enzima ligasa, la cual fue experimentalmente calculada en $0.2\,min^{-1}$

(Cherepanov y de Vries, 2003).

### 6.3.3. Transcripción y traducción del gen CFTR celular y gen CFTR computacional.

El modelado del proceso de transcripción-traducción de genes, o expresión de genes, es un fenómeno bastante estudiado en el campo de biología de sistemas. Actualmente no existe un modelo matemático completo que pueda describirlo de manera global. Esto es debido a que el proceso varía según el organismo (eucariota o procariota), el número de genes implicados en la expresión (desde un gen, hasta redes de genes), el tipo de gen que se quiere modelar, además de que es un proceso totalmente estocástico. Sin embargo, es posible resumir todas estas reacciones bioquímicas en un conjunto pequeño de pasos, lo cual es conveniente para el modelo planteado. Un modelo que provee una buena plataforma teórica para describir la expresión de genes (Elston *et al.*, 2005) se muestra en la Figura 30D).

Basado en este esquema, el control de la transcripción de genes es mediado por factores de transcripción que se unen a elementos promotores, el cual se puede definir mediante la siguiente reacción:

$$\text{Gen CFTR}_{\text{inactivo}} \underset{k_{CGoff}}{\overset{k_{CGon}}{\rightleftharpoons}} \text{Gen CFTR}_{\text{activo}}, \tag{41}$$

donde Gen $\text{CFTR}_{\text{inactivo}}$ define el gen celular CFTR en su estado inactivo (sin expresión) y gen CFTR activo define el mismo gen listo para ser expresado (activo). La transición entre apagado y prendido de un promotor es controlado por las constantes de velocidad de reacción $k_{CGon}$ y $k_{CGoff}$, establecidas en simulación como $10\,min^{-1}$. Una vez activado el gen, este puede transcribirse a moléculas de ARNm. La transcripción se define

mediante la reacción:

$$\text{Gen CFTR}_{\text{activo}} \xrightarrow{k_{CGatc}} \text{ARNm}_{\text{mutado}}, \tag{42}$$

donde $k_{CGatc}$ es la constante cinética del factor de transcripción, establecido como $50\,min^{-1}$. Una vez transcrito, el ARNm puede ser traducido a proteínas. El proceso de traducción de genes se define como:

$$\text{ARNm}_{\text{mutado}} \xrightarrow{k_{CGats}} \text{CFTR}_{\text{mutado}} + \text{ARNm}_{\text{mutado}}, \tag{43}$$

donde $\text{CFTR}_{\text{mutado}}$ son las correspondientes proteínas mutadas y $k_{CGats}$ es la constante de reacción de la traducción, establecida como $0.2\,min^{-1}$. De igual manera, existe el caso donde un gen reprimido se puede activar de una manera rápida, por lo que podría transcribirse a ARNm sin necesidad de haber cambiado de estado (Mads *et al.*, 2005). Este fenómeno se modela mediante la reacción:

$$\text{Gen CFTR}_{\text{inactivo}} \xrightarrow{k_{CGSR}} \text{CFTR}_{\text{mutado}} + \text{Gen CFTR}_{\text{inactivo}}, \tag{44}$$

donde $k_{CGSR}$ es la constante de velocidad de traducción directa, establecida como $5\,min^{-1}$. De igual manera, las reacciones del proceso de transcripción y traducción del gen computacional se pueden definir como:

$$\begin{aligned}
\text{Gen computacional}_{\text{inactivo}} &\underset{k_{CGoff}}{\overset{k_{CGon}}{\rightleftharpoons}} \text{Gen computacional}_{\text{activo}} \\
\text{Gen computacional}_{\text{activo}} &\xrightarrow{k_{atc}} \text{ARNm}_{\text{sano}} \\
\text{ARNm}_{\text{sano}} &\xrightarrow{k_{ats}} \text{CFTR}_{\text{sano}} + \text{ARNm}_{\text{sano}}
\end{aligned}$$

$$\text{Gen computacional}_{\text{inactivo}} \xrightarrow{k_{SR}} \text{CFTR}_{\text{sano}} + \text{Gen computacional}_{\text{inactivo}}. \tag{45}$$

### 6.3.4. Degradación de complejo ADN/ARNm.

La degradación de complejos ADN/ARN es mediada por la enzima RNasa H que se encuentra dentro de las células eucariotas, la cual se adhiere a estos complejos degradando la region de ARN que hibrida con su contraparte en ADN (Fang *et al.*, 2005). Este proceso se lleva a cabo en dos pasos. En el primero de ellos, la enzima se coloca en el complejo (adsorción),

$$\text{RNasa H} + \text{ADN/ARN} \underset{k_d}{\overset{k_a}{\rightleftharpoons}} \text{RNaseH}_{\text{ADN/ARN}}, \tag{46}$$

donde $k_a$ y $k_d$ son las constantes de adsorción y desorción determinadas experimentalmente como $k_a = 3.15\,(\pm 20) \times 10^6\, M^{-1} \cdot s^{-1}$ y $k_d = 0.10\,(\pm 0.05)\, s^{-1}$. La enzima también puede separarse del complejo, movimiento conocido como desorción. Cuando está colocada, la enzima cataliza la reacción degradando la secuencia de ARN,

$$\text{RNaseH}_{\text{ADN/ARN}} \xrightarrow{k_{cat}} \text{Dx} + \text{RNasaH}, \tag{47}$$

donde $k_{cat}$ es la constante de catálisis determinada como $k_{cat} = 0.95\,(\pm 0.10)\, s^{-1}$(Fang *et al.*, 2005). Para modelar la expresión del gen CFTR celular se utilizó este mismo modelo, solamente intercambiando las moléculas ARNm sanas por moléculas ARNm mutadas.

### 6.3.5. Degradación y renovación de ARNm.

Por último, las moléculas de ARNm y proteínas tienen un tiempo promedio de vida antes de ser degradadas. La tasa por la cual estas se degradan esta controlada por las reacciones:

$$\begin{aligned} \text{ARNm}_{\text{sano}} &\xrightarrow{\delta_{hm}} \emptyset \\ \text{CFTR}_{\text{sano}} &\xrightarrow{\delta_{wp}} \emptyset, \end{aligned} \tag{48}$$

donde $\delta_{hm}$ y $\delta_{mp}$ son el tiempo promedio de vida de las moléculas de ARNm y proteínas, respectivamente, establecidas en Mads *et al.* (2005). Similar a los procesos anteriores, intercambiando moléculas de ARNm sano por moléculas de ARNm mutado se obtiene el mecanismo de degradación de moléculas del gen computacional

$$\begin{aligned} \text{ARNm}_{\text{mutado}} &\xrightarrow{\delta_{m}} \emptyset \\ \text{CFTR}_{\text{mutado}} &\xrightarrow{\delta_{mp}} \emptyset. \end{aligned} \tag{49}$$

## 6.4. Simulación del modelo termodinámico de detección y terapia

Basado en el modelo anterior, se generaron 6 escenarios de simulación, tres con el modelo positivo y tres con el modelo negativo. Con la simulación del modelo positivo se buscan 3 objetivos: (1) estimar la eficiencia de detección de moléculas de ARNm mutadas del complejo Dx/Tx, (2) estimar la eficiencia del silenciamiento de la expresión de la proteína aberrante, y (3) predecir la capacidad de generación de tratamiento deseado. En cambio, en el modelo negativo se busca: (1) simular el desempeño de los complejos Dx/Tx para no activarse en ausencia de la mutación, (2) estimar el silenciamiento erróneo de proteínas sanas expresadas, y (3) estimar la generación de

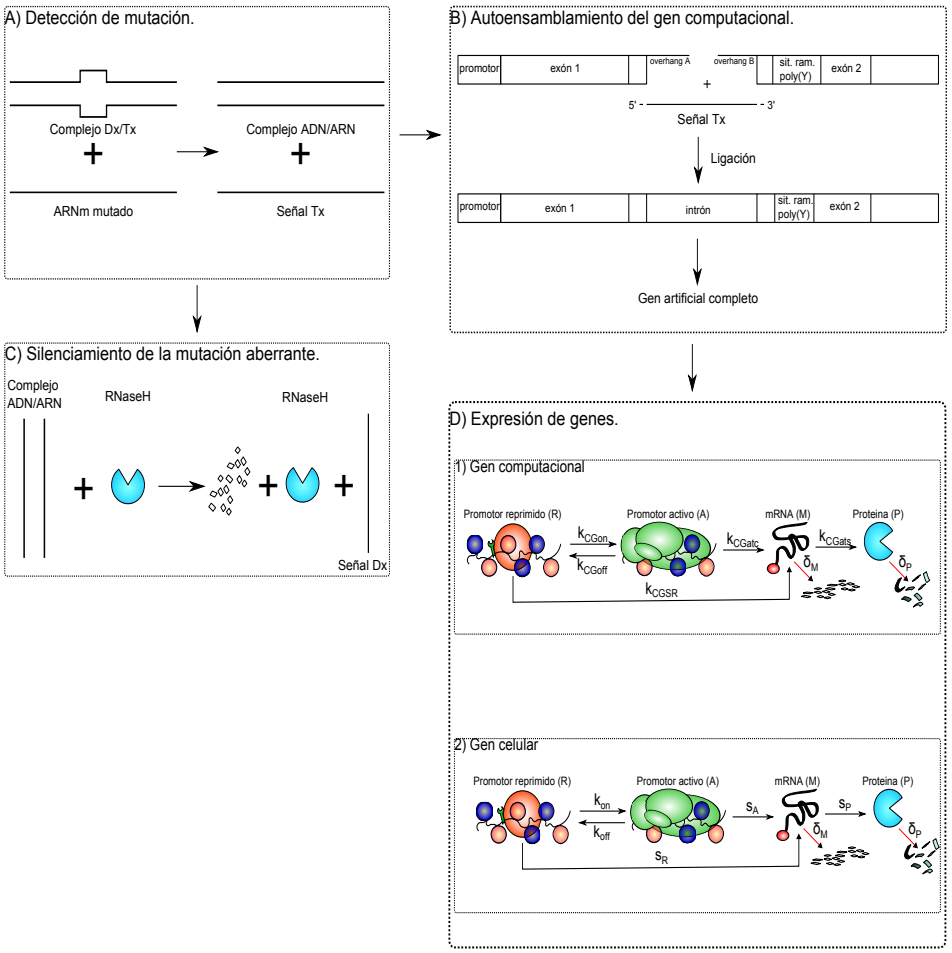



Figura 30: Modelo termodinámico de diagnóstico y terapia (caso positivo). A) Detección de mutación, basado en el modelo termodinámico de diagnóstico (Sección 6.1). B) Autoensamblamiento del gen computacional. El ensamblamiento se activa mediante las señales de tratamiento (Tx), liberadas por el proceso de detección. C) Silenciamiento de la mutación aberrante. La región en ARN del complejo híbrido ADN/ARN sirve como sustrato de la endonucleasa celular RnasaH, silenciando así la expresión del gen mutado. D) Expresión de genes. El tratamiento se lleva a cabo mediante la expresión del gen computacional utilizando el mecanismo celular de transcripción y traducción de genes. Adaptado de , "Stochasticity in gene expression: from theories to phenotypes", De Elston *et al.*, 2005, Nature Reviews. Genetics 6.

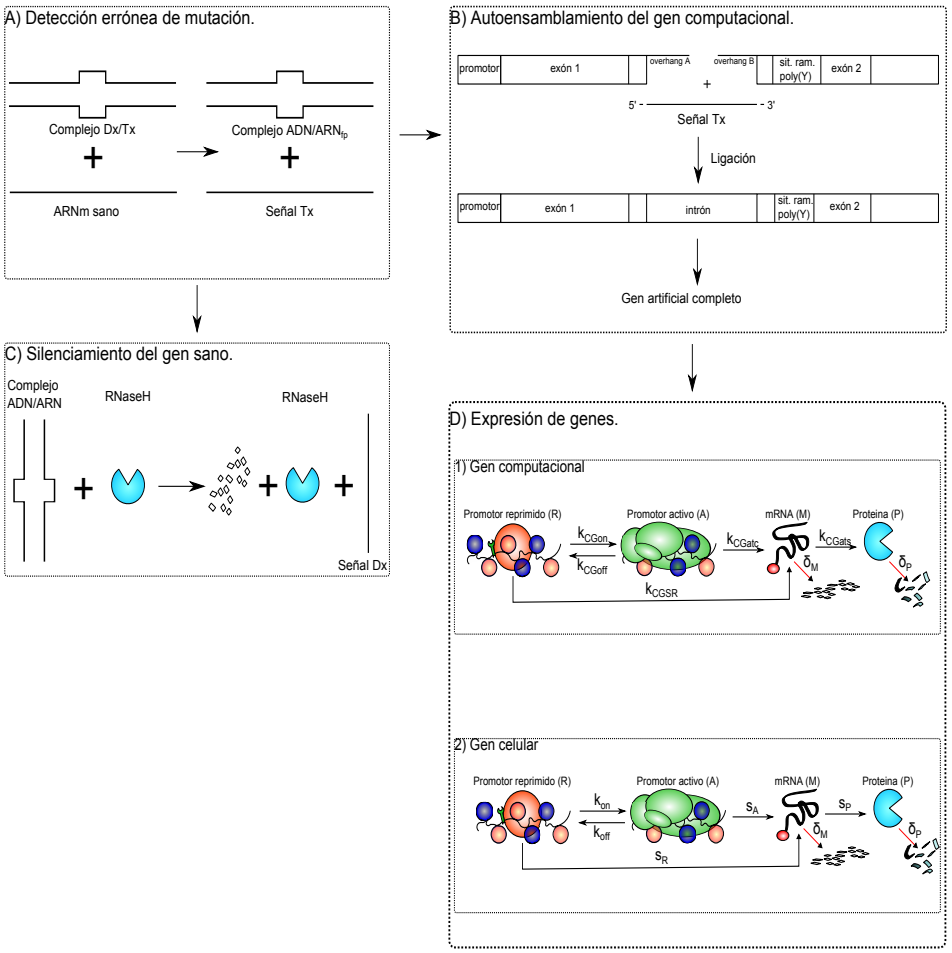



Figura 31: Modelo termodinámico de diagnóstico y terapia (caso negativo). A) Detección errónea de mutación, basado en el modelo termodinámico de diagnóstico negativo (Sección 6.1). B) Auto-ensamblamiento incorrecto del gen computacional. El ensamblamiento se activa mediante las señales de tratamiento (Tx), liberadas por el proceso erróneo de detección y desestabilidad de complejos Dx/Tx. C) Silenciamiento del gen sano. La región en ARN del complejo híbrido ADN/ARN sirve como sustrato de la endonucleasa celular RnasaH, silenciando así la expresión del gen celular. D) Expresión de genes. El tratamiento no deseado se lleva a cabo mediante la expresión del gen computacional utilizando el mecanismo celular de transcripción y traducción de genes. Adaptado de , "Stochasticity in gene expression: from theories to phenotypes", De Elston *et al.*, 2005, Nature Reviews. Genetics 6.

tratamiento no deseado.

En total se realizaron 10 experimentos de manera independiente por modelo, durante 18 horas en tiempo de simulación, realizando muestreo cada segundo. Cada modelo cuenta con un gen celular reprimido y un gen computacional desactivado, como se encontraría de manera hipotética en el núcleo celular. Se estableció la concentración de complejos Dx/Tx en 1,000, 10,000 y 100,000 moléculas. A diferencia de la simulación del modelo de diagnóstico, la concentración de ARNm es variable en el tiempo y dependiente del modelo de transcripción. Los resultados se presentan trazando el número de proteínas generadas por el gen computacional contra el número de proteínas generadas por el gen celular con respecto al tiempo.

### 6.4.1. Modelo de expresión de genes

Empleando el modelo de expresión y traducción de genes (Subsección 6.3.3), se realizó un conjunto de simulaciones para estudiar la expresión de un celular en el tiempo. El objetivo de esta simulación es visualizar la cantidad promedio de moléculas de ARNm transcritas y proteínas traducidas; y de esta manera, utilizar la información generada como referencia comparativa en las simulaciones posteriores. Las figuras 32A) y 32B) muestran el promedio de concentración de moléculas de ARNm y proteínas producidas, respectivamente. En ambos casos, el promedio de concentración de moléculas de ARNm es de 274 moléculas, con una desviación estándar de 20 moléculas; comparado con la concentración de $1077 \pm 142$ proteínas expresadas por el gen celular. Esta diferencia de concentraciones de moléculas de ARNm contra proteínas también se observa en el trabajo de Mads *et al.* (2005), debido principalmente a que se utilizaron las mismas constantes de cinética definidas por los autores.

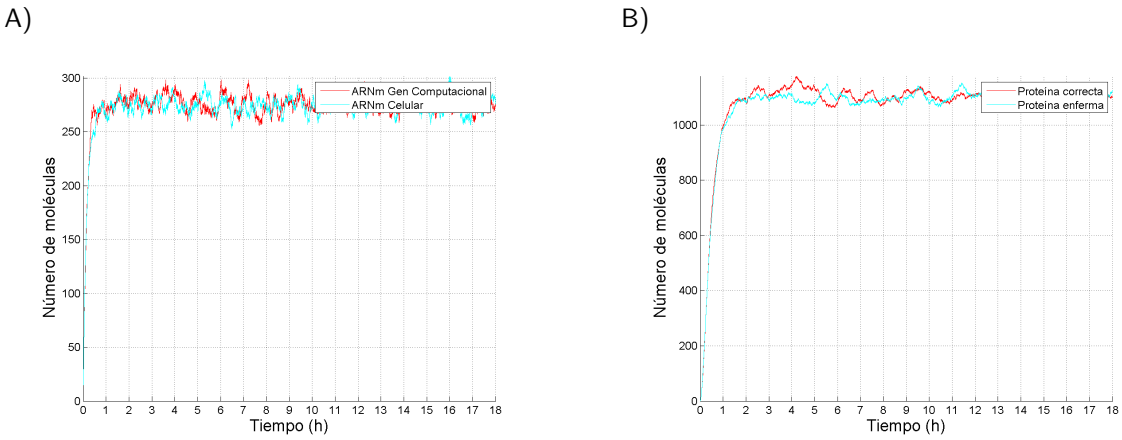



Figura 32: Resultados de simulación del modelo de expresión de genes. A) Transcripción de genes. B) Traducción de genes.

### 6.4.2. Complejo tipo I

Utilizando el mismo esquema del modelo de diagnóstico, las simulaciones se dividieron en tres escenarios, tanto para el modelo positivo como para el modelo negativo. En el primero de ellos se consideró un total de 1,000 moléculas de complejos Dx/Tx. En el caso del modelo positivo (Figura 33A), se puede apreciar que efectivamente la señal de diagnóstico del complejo Dx/Tx interfiere con la síntesis de proteínas aberrantes expresadas por el gen celular, manteniéndose este número constante en alrededor de $370 \pm 7$ proteínas. A su vez, junto con la activación del gen computacional, expresándose en promedio $169 \pm 191$ proteínas sanas. Sin embargo, debido a la escasa concentración inicial de complejos Dx/Tx, no se logra suprimir por completo la expresión de proteínas aberrantes.

En el segundo escenario, las simulaciones se realizaron con una concentración de 10,000 complejos Dx/Tx, como se muestra en la Figura 33C), en el modelo positivo, se logró reducir la síntesis de proteínas aberrantes a un número aproximado de $57 \pm 2$ proteínas enfermas. Por otro lado, la activación del gen computacional en el tiempo

Tabla 10: Expresión relativa de proteínas en modelo negativo (complejo tipo I). La expresión relativa se estimó con la diferencia entre la proteína total generada en el modelo (celular + gen computacional) y aquella generada por el modelo de transcripción y traducción celular (Figura 32B)).

| Número de complejos Dx/Tx | Proteína celular | Proteína gen computacional | Total | % de expresión relativa |
|---|---|---|---|---|
| 1,000 | $941 \pm 17$ | $331 \pm 175$ | 1464 | $+20\,\%$ |
| 10,000 | $748 \pm 12$ | $112 \pm 219.82$ | 1093 | $-10\,\%$ |
| 100,000 | $456 \pm 7$ | $121 \pm 196$ | 760 | $-38\,\%$ |

permite la expresión de proteínas sanas, en un promedio de $6 \pm 71$ proteínas sanas.

Por último, el tercer escenario se realizó utilizando una concentración inicial de 100,000 complejos Dx/Tx. Los resultados obtenidos para el modelo positivo (Figura 33E) muestran que es posible suprimir casi en su totalidad el número de proteínas aberrantes, permitiendo además la activación del gen computacional en menor tiempo, comparado con el escenario anterior. Este hecho sugiere que una concentración adecuada de complejos Dx/Tx podría reemplazar totalmente la cantidad de proteína aberrante por proteína funcional en la membrana celular, recuperando así la capacidad de transportar iones de cloro, lo que sería un claro indicio terapéutico para el paciente.

Aunque en general las simulaciones arrojaron buenos comportamientos para el modelo positivo, no se obtuvo el comportamiendo esperado en el modelo negativo. Por ejemplo, las figuras 33B), 33D) y 33F) sugieren que los complejos Dx/Tx interfieren poco con las moléculas de ARNm funcionales, por lo que la expresión de proteínas se llevaría a cabo de manera natural. Sin embargo, el auto-ensamblamiento del gen computacional sí se realiza, debido a la separación espontánea de los complejos Dx/Tx, produciéndose como consecuencia proteína CFTR adicional.

Este comportamiento se debe a la diferencia entre concentraciones de sondas y genes computacionales desactivados. Como se puede observar en la Figura 24, los complejos

Dx/Tx no son totalmente estables, por lo que el número de moléculas de tratamiento (Tx) y diagnóstico (Dx) libres tiende a crecer conforme se aumenta la concentración inicial de complejos, lo cual provoca el auto-ensamblamiento del gen computacional y su posterior expresión. Cabe mencionar sin embargo que, la proteína expresada por el gen computacional es funcionalmente idéntica a la proteína expresada por la célula, por lo que la célula en teoría no se vería afectada, aunque la sobre-expresión de la misma (Tabla 10) podría llegar a ser contraproducente, por lo que es necesario un futuro análisis.

### 6.4.3. Complejo tipo II

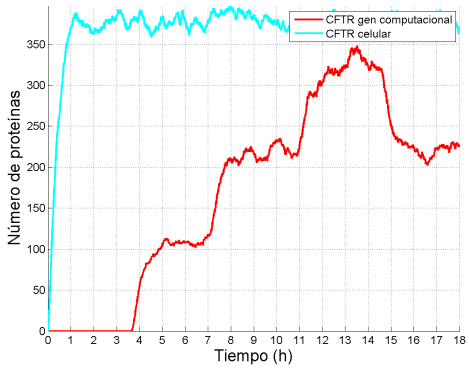
Se consideran los mismos tres escenarios de simulación. Los resultados obtenidos en el modelo positivo fueron los siguientes: en el primer escenario (1,000 moléculas de complejos Dx/Tx) se puede observar que al igual que los resultados del complejo anterior, la cantidad de sondas no es suficiente para inhibir la expresión de proteínas aberrantes (Figura 34A), lográndose expresar un promedio de $372 \pm 6.5$ proteínas, junto con una baja expresión por parte del gen computacional ($34 \pm 154$).

En el segundo escenario se llevaron a cabo las simulaciones con una concentración inicial de 10,000 moléculas de complejos Dx/Tx, lográndose inhibir la expresión de proteínas aberrantes considerablemente (Figura 34C), esto es $59 \pm 2$ proteínas disfuncionales y una expresión promedio por parte del gen computacional de $128 \pm 146$ proteínas. Debido a esta gran diferencia, existe una alta probabilidad de que las proteínas sintetizadas por el gen computacional se depositen en la membrana celular, con lo cual se generaría un tratamiento al paciente.

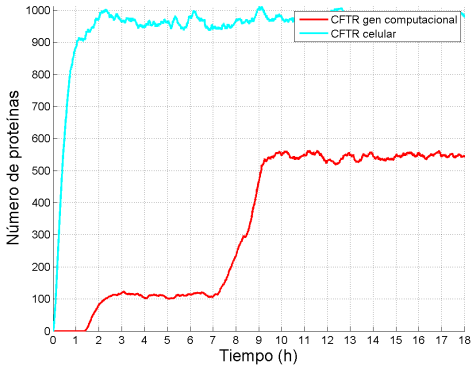
Por otro lado, las simulaciones del tercer escenario (con una concentración inicial de 100,000 moléculas de complejos) lograron inhibir en su totalidad la expresión

1,000 moléculas.

A) Modelo positivo

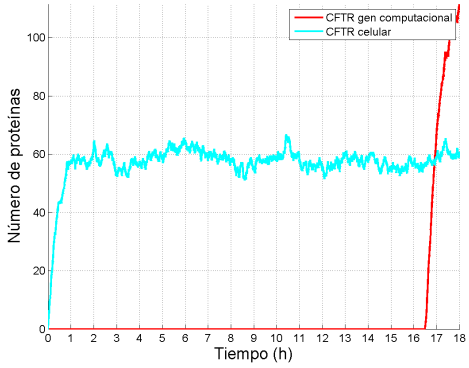



B) Modelo negativo

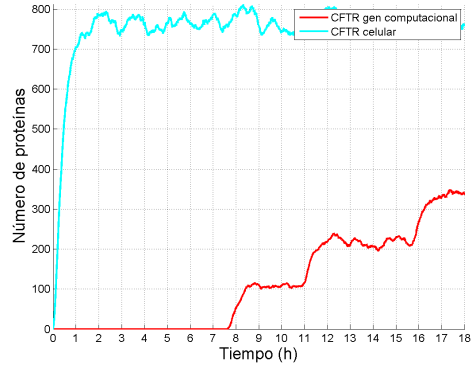



10,000 moléculas.

C) Modelo positivo

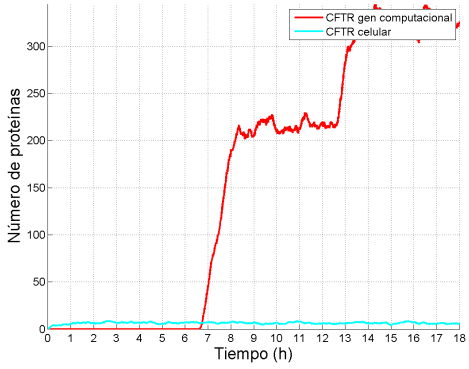



D) Modelo negativo




100,000 moléculas.

E) Modelo positivo




F) Modelo negativo

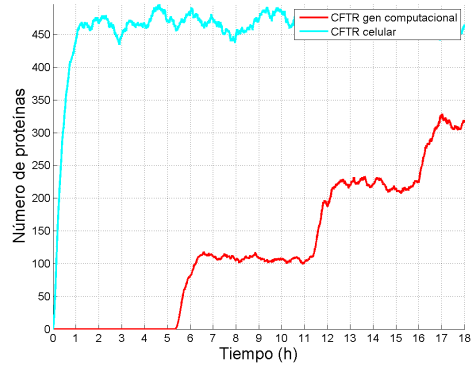



Figura 33: Resultados de simulación: modelo de diagnóstico y tratamiento (complejo tipo I).

de proteínas aberrantes, expresándose solamente las proteínas del gen computacional (Figura 34E), por lo que la célula recuperaría la función del canal de iones de cloro, beneficiando al paciente.

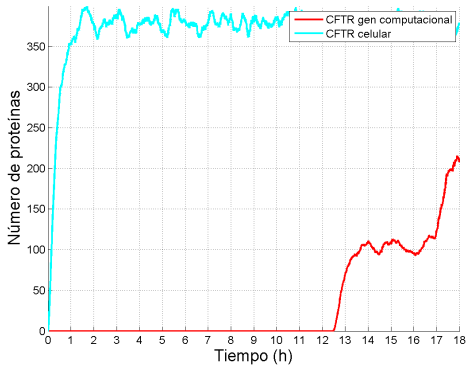
En el caso del modelo negativo, se puede observar un comportamiento similar al obtenido por el modelo negativo del complejo anterior en cada uno de los escenarios. Esto es, se puede apreciar la activación del gen computacional por la inestabilidad de los complejos Dx/Tx asi como la inhibición del ARNm sano (figuras 34B) y 34F). La excepción es el escenario 2, que aunque concuerda con las simulaciones de la configuración anterior, el gen celular no es inhibido cuantitativamente, y además ocurre un auto-ensamblamiento tardío del gen computacional (Figura 34D). A simple vista, esto indica que la concentración óptima de complejos Dx/Tx para un funcionamiento correcto del gen computacional se encuentra en el rango de 10,000 moléculas, no obstante sería necesario realizar un análisis más detallado.

### 6.4.4. Complejo tipo III

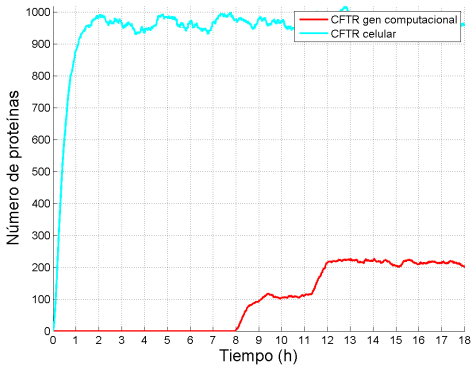
Las simulaciones se presentan en tres escenarios diferenciados por las concentraciones iniciales de complejos Dx/Tx. En el primer escenario se consideró un total de 1,000 moléculas de complejos representados por una configuración tipo III, las cuales, en el caso del modelo positivo (Figura 35A), lograron inhibir la expresión de proteínas aberrantes hasta un 35 % aproximadamente ($\approx 338 \pm 6.2$ proteínas aberrantes), mientras el gen computacional logró expresarse hasta un 32 % ($\approx 102 \pm 206$ proteínas del gen computacional) comparado con los datos obtenidos en las simulaciones del modelo de expresión de genes ($\approx 1,077$ proteínas), demostrando así un mejor desempeño que los complejos anteriores (Figura 32B). No obstante, los complejos no tuvieron el mismo rendimiento en su contraparte negativo (Figura 35B), ya que lograron inhibir la ex-

1,000 moléculas.

A) Modelo Positivo

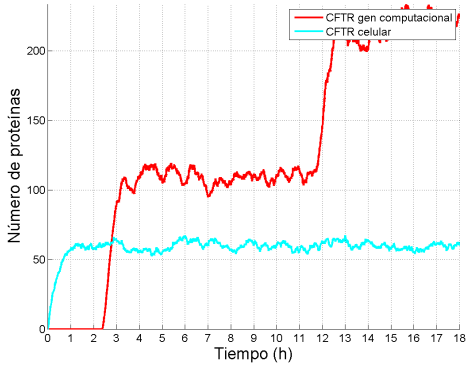



B) Modelo Negativo

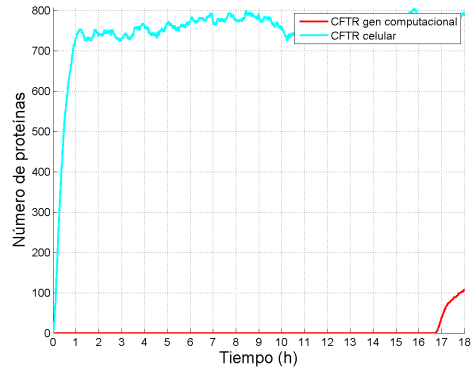



10,000 moléculas.

C) Modelo Positivo

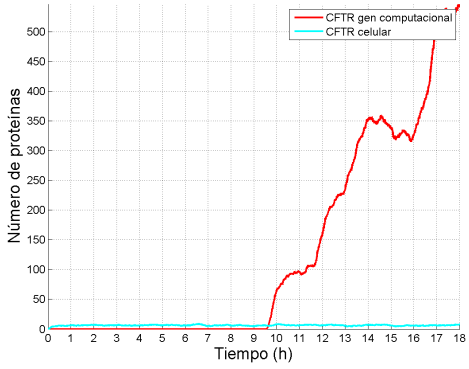



D) Modelo Negativo




100,000 moléculas.

E) Modelo Positivo




F) Modelo Negativo

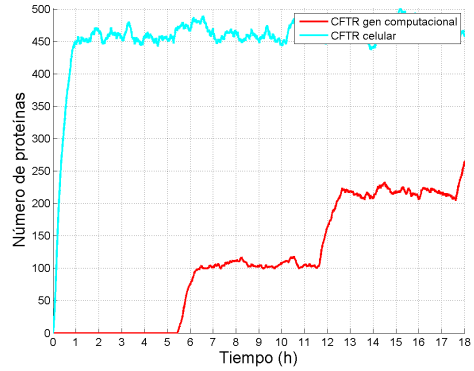



Figura 34: Resultados de simulación: modelo de diagnóstico y tratamiento (complejo tipo II).

presión de proteínas celulares sanas hasta un 35 % ($\approx 368 \pm 6$ moléculas), además del auto-ensamblamiento no deseado del gen computacional ($\approx 124 \pm 191$ moléculas).

En el segundo escenario, se utilizó una concentración inicial de 10,000 moléculas de complejos Dx/Tx en las simulaciones, las cuales para el modelo positivo (Figura 35C) lograron inhibir la expresión de proteínas aberrantes hasta en un 5 % aproximadamente ($\approx 51 \pm 2$ moléculas) mientras que la expresión de proteínas sanas por parte del gen computacional alcanzó un 45 % ($\approx 283 \pm 207$ moléculas) del rendimiento esperado de acuerdo al modelo de expresión de genes (Figura 32). Sin embargo, en el caso del modelo negativo (Figura 35D), el número de proteínas sanas decayó hasta en un 5.5 % ($\approx 57 \pm 2$ moléculas) por inhibición, mientras que la expresión del gen computacional alcanzó un 34 % ($\approx 234 \pm 135$ moléculas). Si a esto le agregamos el lento auto-ensamblamiento del gen computacional, resulta en una baja concentración de proteínas CFTR sanas, lo que podría ser contraproducente para el paciente.

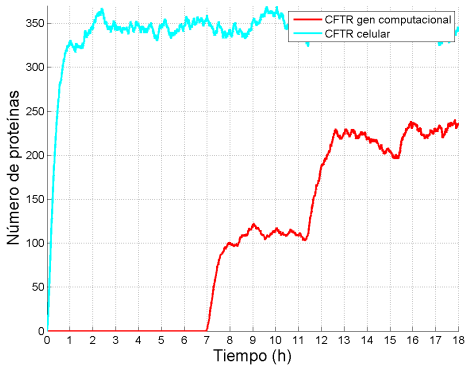
Por último, el tercer escenario consideró una concentración inicial de 100,000 complejos, lo que permite una total inhibición de proteínas aberrantes ($\approx 5 \pm 0.5$ moléculas), mientras se logra una expresión de aproximadamente $223 \pm 190$ proteínas (38 %) del gen computacional en el modelo positivo (Figura 35E). Por su parte, el modelo negativo obtuvo un comportamiento similar al modelo positivo, en el que se alcanza una total inhibición de proteínas sanas ($\approx 7 \pm 0.6$ moléculas), contra una expresión de alrededor de $224 \pm 100$ moléculas de proteínas, esto es, 30 % del rendimiento esperado por parte del gen computacional (Figura 35F).

### 6.4.5. Análisis de resultados

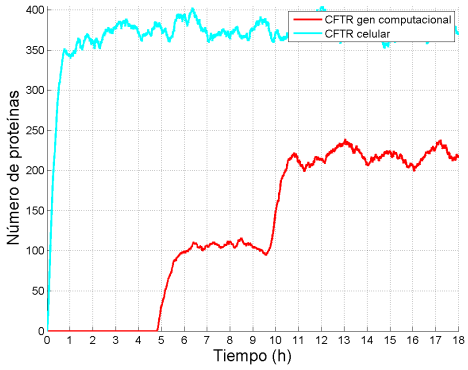
Comparando los resultados de simulación obtenidos por cada configuración, se puede observar que, sin excepción, las configuraciones en el modelo positivo lograron diag-

1,000 moléculas.

A) Modelo positivo

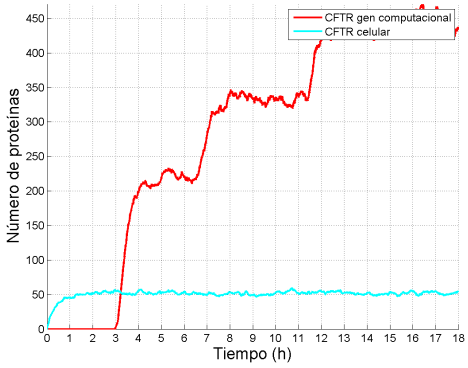



B) Modelo negativo

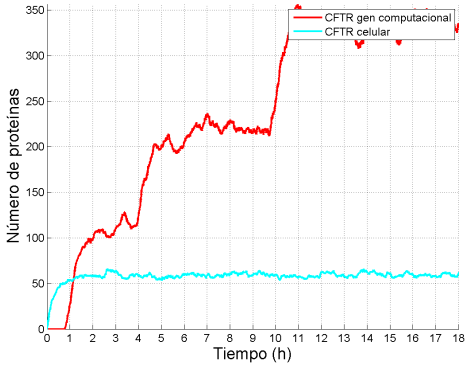



10,000 moléculas.

C) Modelo positivo

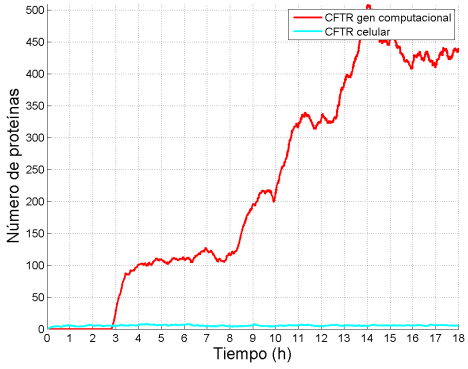



D) Modelo negativo




100,000 moléculas.

E) Modelo positivo




F) Modelo negativo

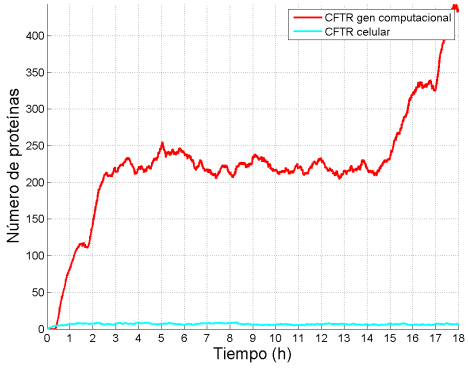



Figura 35: Resultados de simulación modelo de diagnóstico y tratamiento (complejo tipo III).

nosticar la mutación, inhibiendo la expresión de proteínas aberrantes y generaron un tratamiento mediante la expresión de proteínas CFTR sanas (Figura 36). Por otro lado, se confirma el comportamiento presentado en las simulaciones del modelo de diagnóstico, en el que a mayor número de complejos Dx/Tx, mayor la inhibición de proteínas disfuncionales. Sin embargo, esto no significa un mayor número de proteínas expresadas por el gen computacional, como se observa con la configuración tipo III, donde la expresión aumenta (en promedio) del escenario 1 (1,000 moléculas Dx/Tx) al escenario 2 (10,000 moléculas Dx/Tx), pero en el tercer escenario (100,000 moléculas) la expresión vuelve a disminuir. Un comportamiento parecido presenta la configuración tipo II, donde al aumentar la concentración de complejos de 1,000 a 10,000 moléculas, la expresión de proteínas por parte del gen computacional aumenta, pero al aumentar la concentración a 100,000 complejos, el número de proteínas disminuye considerablemente. De igual manera, la configuración tipo I presenta un comportamiento parecido al de las configuraciones anteriores, donde se observa una expresión de proteínas por parte del gen computacional fluctuante independiente del número de moléculas de complejos Dx/Tx.

Parte de este comportamiento fluctuante entre configuraciones, se piensa, es debido al lento ensamblamiento del gen computacional atribuido a la diferencia del número de moléculas entre el gen computacional y los complejos. Como se recordará, las simulaciones se realizaron utilizando un solo gen computacional desactivado, el cual se ensambla mediante la señal de tratamiento Tx, y la concentración de esta señal depende de la estabilidad de los complejos Dx/Tx y la presencia de moléculas ARNm. Además, las señales de diagnóstico también se liberan cuando un complejo ADN/ARN es degradado mediante de la enzima ARNasa H. Esto quiere decir que, en cualquier momento, existe un número de señales de diagnóstico capaces de aparearse ya sea con una señal

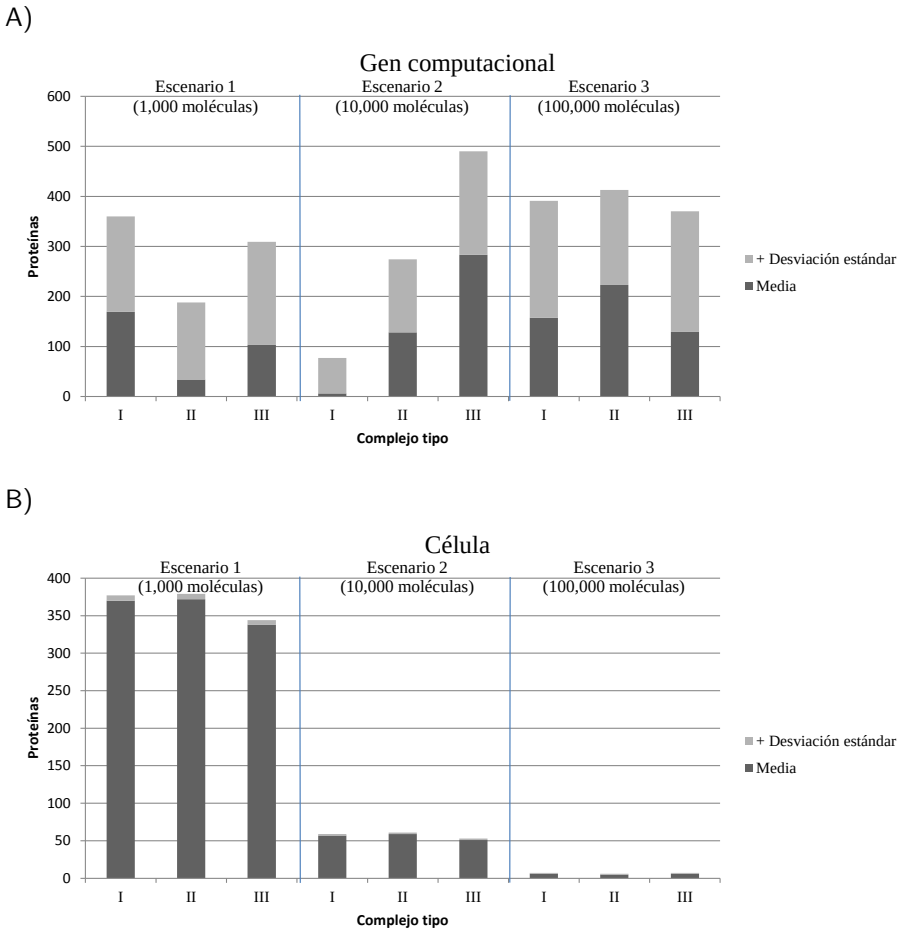



Figura 36: Síntesis de proteínas en el modelo de diagnóstico y tratamiento (caso positivo). A) Proteínas expresadas por el gen computacional. B) Proteínas expresadas por el gen celular.

de tratamiento (Tx) o una molécula de ARNm mutada libre; lo que se traduce en una menor probabilidad de que el gen computacional sea ensamblado por una señal Tx.

Otra observación interesante que arrojaron las simulaciones, es el auto-ensamblamiento del gen computacional en el modelo negativo (Figura 37), el cual es expresado de la misma manera fluctuante que en el modelo positivo, pero a diferencia de este, al aumentar el número de complejos Dx/Tx de 10,000 a 100,000 moléculas, la expresión del gen computacional en la configuración II aumenta, suceso que no ocurre en las configuraciones restantes.

Por otro lado, los resultados del modelo negativo también confirman el mal comportamiento del complejo tipo III observado en la simulación del modelo negativo de diagnóstico, al inhibir la expresión de proteínas sanas en la célula, la cual aumenta con la concentración de complejos Dx/Tx.

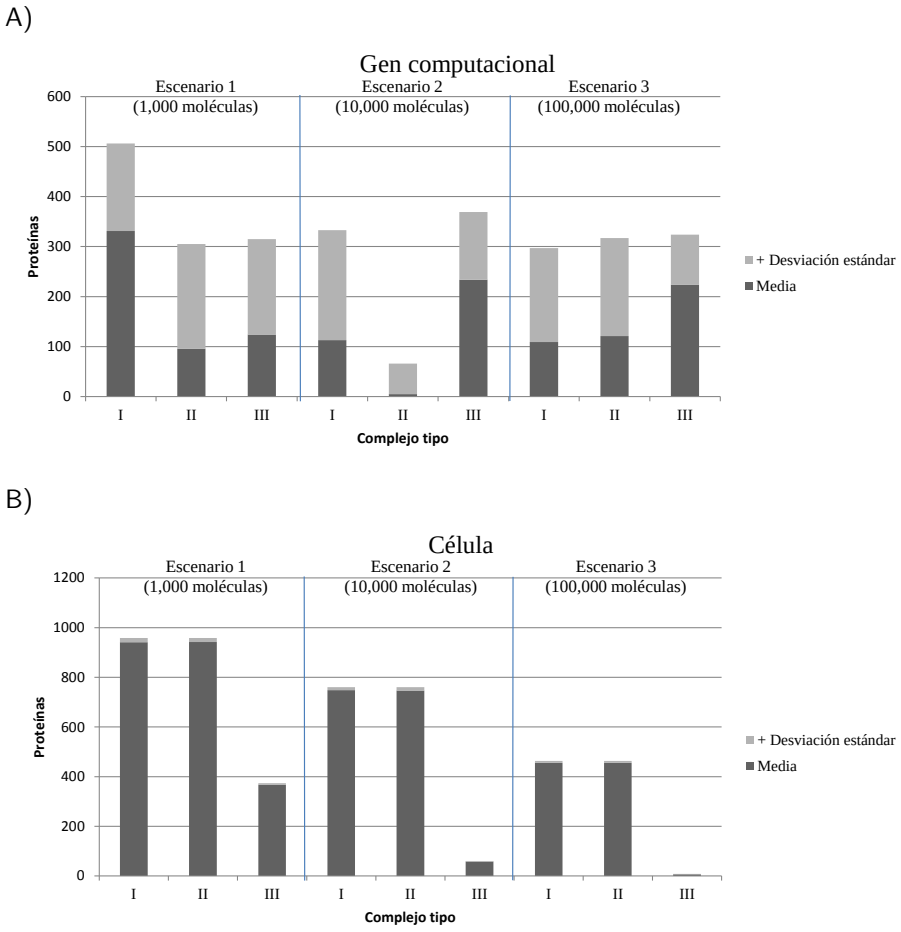



Figura 37: Síntesis de proteínas en el modelo de diagnóstico y tratamiento (caso negativo). A) Proteínas expresadas por el gen computacional. B) Proteínas expresadas por el gen celular.

Aunque en general todas las configuraciones tuvieron un mal comportamiento, en el modelo negativo, cabe mencionar que la proteína expresada por el gen computacional es idéntica, funcionalmente hablando, a la expresada por la célula, por lo que en caso de inhibirse las proteínas de la célula, estas serían reemplazadas por las del gen

computacional. Sin embargo, es posible disminuir este problema al variar la concentraciones de complejos Dx/Tx, la cual observando los datos de simulación, se piensa que la concentración óptima sea de alrededor de 10,000 moléculas.

A nivel de escenarios, se deduce que la mejor configuración para una concentración de 1,000 moléculas fue la tipo I. Aunque la configuración tipo II logró inhibir un mayor número de proteínas aberrantes en el modelo positivo, inhibir una menor cantidad de proteínas sanas en el modelo negativo y mantener el gen computacional desactivado por más tiempo, la configuración tipo I logró expresar una mayor cantidad de proteínas sanas (gen computacional) en el modelo positivo. Sin embargo, esto no ocurre en el caso con 10,000 moléculas de complejos, donde el complejo ganador fue el tipo II. Como se puede observar en la Figura 36, esta configuración expresó la mayor cantidad de proteínas sanas debido al ensamblamiento temprano del gen computacional, comparado con la configuración tipo I. De manera similar, comparando la configuración tipo II con la tipo III, esta última obtiene un mayor número de proteínas sanas expresadas por el gen computacional, junto con un mayor número de proteínas aberrantes inhibidas, pero al comparar los resultados del modelo negativo, la configuración tipo III obtiene el peor desempeño. Por último, la configuración que logró un mejor desempeño con 100,000 moléculas de complejos fue de nuevo el complejo tipo II, con una mayor cantidad de proteínas expresadas por el gen computacional y proteínas aberrantes inhibidas en el modelo positivo; además de un buen desempeño en el modelo negativo. En conclusión, la mejor combinación tipo complejo/concentración es la tipo II, con una concentración de 100,000 moléculas debido a que obtuvo una especificidad y sensibilidad del 41.37 % y 98.8 %, respectivamente en el modelo de diagnóstico, además de una estabilidad del 99.75 % y un buen desempeño en el modelo de diagnóstico y tratamiento.

## Conclusiones

Este trabajo presenta un modelo de optimización para el diseño de sondas en genes computacionales utilizando algoritmos genéticos. Como caso de estudio, se diseñó un autómata que fuera capaz de detectar la deleción $\Delta F508$ del gen de la fibrosis quística y de proveer un tratamiendo de ser necesario. Los diseños se verificaron mediante simulación estocástica utilizando parámetros y condiciones fisiológicas. Inicialmente se describió el problema del diseño de sondas para el diagnóstico de mutaciones aberrantes, en donde se busca una configuración de complejos Diagnóstico/Tratamiento, tal que se maximice el número de verdaderos positivos (detección correcta) mientras se minimiza el número de falsos negativos (detección errónea). Para esto se definieron varias funciones a optimizar, tales como la estabilidad en complejos Dx/Tx, la estabilidad en complejos Dx/ARNm, constantes de cinética, etc.

Posteriormente, se desarrollaron los componentes utilizados en el algoritmo genético, que incluyó una representación capaz de modelar estructuras secundarias de ácidos nucleicos, además de operadores de cruzamiento y mutación que acepten la representación. El algoritmo se implementó sobre la plataforma de optimización OPT4J, agregando las nuevas representaciones y operadores al mismo. La selección de configuraciones óptimas de parámetros se realizó mediante el uso de métricas para el desempeño de algoritmos evolutivos multi-objetivo, particularmente la métrica de cubrimiento desarrollada por Zitzler (1999).

Por último, se describieron los modelos utilizados en la simulación. Se comienza con el modelo de detección, con el cual se estudió el comportamiento hipotético de los complejos Dx/Tx en un ambiente de laboratorio, principalmente para estimar su

capacidad de diagnóstico. Después se describió el modelo de detección y tratamiento, el cual estima el desempeño de los complejos acoplados con el gen computacional en un ambiente interno celular. La simulación se realizó utilizando el algoritmo de Gillespie (Wilkinson (2006)), implementado en el software de simulación Dizzy (Ramsey *et al.* (2005)). Analizando los resultados obtenidos en el modelo de diagnóstico, se comprobó el desempeño de los complejos Dx/Tx diseñados, lográndose obtener una especificidad y sensibilidad del 41.37 % y 98.8 %, respectivamente, junto con una estabilidad del 99.75 % en una concentración de 100,000 moléculas (complejo tipo II). A su vez, en la simulación del modelo de diagnóstico y tratamiento positivo se logró inhibir en su totalidad la expresión de proteínas aberrantes, mientras que la expresión del gen computacional se lleva a cabo. Sin embargo, los resultados muestran que la inhibición de proteínas sanas celulares en el modelo de diagnóstico y tratamiento negativo también es posible, aunque este efecto se compensa un poco con el auto-ensamblamiento del gen computacional, que al expresarse de manera paralela con el gen natural, ayuda a reemplazar la proteína inhibida, con la probable sobre-expresión del mismo.

A pesar del avance realizado, aún existe trabajo por hacer, tanto en la optimización de complejos Dx/Tx como en la simulación. En el primer caso, resultaría interesante evaluar la aptitud de los individuos utilizando el número de verdaderos positivos y falsos positivos generados en simulación por la configuración del complejo. También se propone agregar más operadores de cruzamiento y mutación, hasta un nuevo esquema de inicialización de individuos. En cuanto a la simulación, un factor por explotar es encontrar un compromiso entre concentraciones de complejos y ARNm que permita mejorar los parámetros de especificidad y sensibilidad. Por otro lado, se puede realizar una mejora en los resultados de simulación, principalmente en la aceleración del auto-ensamblamiento del gen computacional, realizando mejoras tanto al modelo, como en la

implementación de nuevos modelos de simulación mediante algoritmos diferentes (e.g. simulación estocástica espacial).

Pese a que la implementación del gen computacional a nivel celular en un futuro es incierto, el diseño y la aplicación de complejos optimizados para el diagnóstico de enfermedades producidas por mutaciones es factible con la tecnología de hoy en día. Sin embargo, debido a los alcances de esta investigación, resultó imposible realizar experimentos en laboratorio para corroborar el desempeño de los complejos, por lo que se propone como trabajo a futuro.

## Referencias bibliográficas

Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science*, **266**: 1021–1024.

Adleman, L. M. (1996). On constructing a molecular computer. *DIMACS*, **27**(1): 1–21.

Benenson, Y., Paz-Elizur, T., Adar, R., Keinan, E., Livneh, Z., y Shapiro, E. (2001). Programmable and autonomous computing machine made of biomolecules. *Nature*, **414**(6862): 430–434.

Benenson, Y., Paz-Ellizur, T., Livneh, Z., y Shapiro, E. (2003). DNA molecule provides a computing machine with both data and fuel. *Proceedings of the National Academy of Sciences*, (100): 2191–2196.

Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., y Shapiro, E. (2004). An autonomous molecular computer for logical control of gene expression. *Nature*, **429**(6990): 423–429.

Cherepanov, A. V. y de Vries, S. (2003). Kinetics and thermodynamics of nick sealing by T4 DNA ligase. *European Journal of Biochemistry*, **270**(21): 4315–4325.

Coello, C. A. C., Lamont, G. B., y Veldhuizen, D. A. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, segunda edición.

Coello-Coello, C., Dhaenens, C., y Jourdan, L. (2010). Multi-objective combinatorial optimization: Problematic and context. *C. Coello Coello, C. Dhaenens, y L. Jourdan, (Eds.) Advances in Multi-Objective Nature Inspired Computing*, Vol. 272 de *Studies in Computational Intelligence*, páginas 1–21. Springer Berlin / Heidelberg. ISBN: 978-3-642-11217-1. doi: 10.1007/978-3-642-11218-8_1.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John

Wiley & Sons, Chichester.

Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**(2): 182–197.

Elston, T. C., Blake, W. J., y Collins, J. J. (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews. Genetics*, **6**(6): 451–464.

Emmerich, M., Beume, N., y Naujoks, B. (2005). An emo algorithm using the hypervolume measure as selection criterion. *Proceedings of the Third international conference on Evolutionary Multi-Criterion Optimization*, EMO'05, páginas 62–76, Berlin, Heidelberg. Springer-Verlag. ISBN: 3-540-24983-4, 978-3-540-24983-2. doi: 10.1007/978-3-540-31880-4_5.

Fang, S., Lee, H. J., Wark, A. W., Kim, H. M., y Corn, R. M. (2005). Determination of ribonuclease h surface enzyme kinetics by surface plasmon resonance imaging and surface plasmon fluorescence spectroscopy. *Analytical Chemistry*, **77**(20): 6528–6534.

Farhang-Mehr, A. y Azarm, S. (2002). Diversity assessment of pareto optimal solution sets: an entropy approach. *Proceedings of the World on Congress on Computational Intelligence*, Vol. 1, páginas 723–728, Los Alamitos, CA, USA. IEEE Computer Society. ISBN: 0-7803-7282-4. doi: http://doi.ieeecomputersociety.org/10.1109/CEC.2002.1007015.

Feynman, R. P. (1961). There's plenty of room at the bottom: An in vitation to enter a new field of physics. *D. Gilbert Editor Miniaturization*, páginas 282–296, Reinhold, New York.

Flintoft, L. (2010). Cellular defence: Human cells clear foreign DNA. *Nature Reviews Genetics*, **11**(3): 172–172.

Fonseca, C. y Fleming, P. (1993). Genetic algorithms for multiobjective optimization:

Formulation, discussion and generalization. *Proceedings of the 5th International Conference on Genetic Algorithms*, páginas 416–423, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Gibbons, A., Amos, M., y Hodgson, D. (1996). Models of DNA computation. *W. Penczek y A. Szalas, (Eds.) Mathematical Foundations of Computer Science 1996*, Vol. 1113 de *Lecture Notes in Computer Science*, páginas 18–36. Springer Berlin/Heidelberg. ISBN: 978-3-540-61550-7. 10.1007/3-540-61550-4_138.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, **81**(25): 2340–2361.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman, Boston, MA. ISBN: 0201157675.

Gordon, P. M. K. y Sensen, C. W. (2004). Osprey: A comprehensive tool employing novel methods for the design of oligonucleotides for DNA sequencing in microarrays. *Nucleic Acids Res.*, **32**(17): e133.

Graugnard, E., Cox, A., Lee, J., Jorcyk, C., Yurke, B., y Hughes, W. (2010). Kinetics of DNA and RNA hybridization in Serum and Serum-SDS. *IEEE Transactions on Nanotechnology*, **9**(5): 603 –609.

Hansen, M. P. (1998). *Metaheuristics for multiple objective combinatorial optimization*. Tesis de doctorado, Institute of Mathematical Modelling, Technical University of Denmark.

Hartamis, J. (1995). On the weight of computations. *Bulletin of the European Association for Theorical Computer Science*, **55**: 136–138.

Head, T., Chen, X., Nichols, N. J., Yamamura, N., y Gal, S. (2002). Aqueuos solutions of algorithmic problems emphasizing knights on a 3x3.. *DNA 7*, páginas 191–202.

Henkel, C. V., Bladergroen, R. S., Ralog, C. I., Deelder, A. M., y Head, T. (2005).

Protein output for DNA computing. *Natural Computing*, **4**: 1–10.

Hertz, A., Jaumard, B., Ribeiro, C. C., y Filho, W. P. F. (1994). A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. *RAIRO - Operations Research - Recherche Opérationnelle*, **28**: 303–328.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA. ISBN: 0-262-58111-6.

Ignatova, Z., Martínez-Pérez, I., y Zimmermann, K. (2008). *DNA Computing Models*. Advances in information security. Springer. ISBN: 9780387736372.

Ishibuchi, H. y Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **28**(3): 392 –403.

Knowles, J. y Corne, D. (2002). On metrics for comparing nondominated sets. *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, Vol. 1, páginas 711 –716. doi: 10.1109/CEC.2002.1007013.

Kuramochi, J. y Sakakibara, Y. (2006). Intensive *In Vitro* experiments of implementing and executing finite automata in test tube. *A. Carbone y N. Pierce, (Eds.) DNA Computing*, Vol. 3892 de *Lecture Notes in Computer Science*, páginas 193–202. Springer Berlin. ISBN: 978-3-540-34161-1. doi: 10.1007/11753681_15.

Lalkhen, A. G. y McCluskey, A. (2008). Clinical tests: sensitivity and specificity. *Continuing Education in Anaesthesia, Critical Care & Pain*, **8**(6): 221–223.

Lipton, R. J. (1995). DNA solution of hard computational problems. *Science*, **268**: 542–545.

Lukasiewycz, M., Glaß, M., Reimann, F., y Teich, J. (2011). Opt4J - A Modular Framework for Meta-heuristic Optimization. *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 2011)*, páginas 1723–1730, Dublin.

Mads, K., C., E. T., Blake, W. J., y J., C. J. (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, **6**: 451–464.

Martínez-Pérez, I. M. (2007). *Biomolecular Computing Models for Graph Problems and Finite State Automata*. Tesis de doctorado, Technische Universität Hamburg.

Martínez-Pérez, I. M., Zhang, G., Ignatova, Z., y Zimmermann, K.-H. (2007). Computational genes: a tool for molecular diagnosis and therapy of aberrant mutational phenotype. *BMC Bioinformatics*, **8**(1): 365.

Martínez-Pérez, I. M., Karl, H. Z., y Ignatova, Z. (2009). An autonomous DNA model for finite state automata. international journal of bioinformatics research and applications. *International Journal of Bioinformatics Research and Applications*, **5**(1): 81–96.

Meunier, H., Talbi, E.-G., y Reininger, P. (2000). A multiobjective genetic algorithm for radio network optimization. *Proceedings of the 2000 Congress on Evolutionary Computation*, Vol. 1, páginas 317 –324. doi: 10.1109/CEC.2000.870312.

Mühlenbein, H. (1997). The equation for response to selection and its use for prediction. *Evolutionary Computation*, **5**(3): 303–346.

Nakagawa, H., Sakamoto, K., y Sakakibara, Y. (2006). Development of an *In Vivo* computer based on *Escherichia coli. A. Carbone y N. Pierce, (Eds.) DNA Computing*, Vol. 3892 de *Lecture Notes in Computer Science*, páginas 203–212. Springer Berlin / Heidelberg. ISBN: 978-3-540-34161-1. doi: 10.1007/11753681_16.

Ouyang, Q., Kaplan, P. D., Liu, S., y Libchaber, A. (1997). DNA solution of the maximal clique problem. *Science*, **278**(5337): 446–449.

Phillips, A. y Luca, C. (2009). A programming language for composable DNA circuits. *Journal of The Royal Society Interface*, **6**(4): S419–S436.

Ramsey, S., Orrell, D., y Bolouri, H. (2005). Dizzy: stochastic simulation of large-scale

genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, **3**(2): 415–436.

Riordan, J. R., Rommens, J. M., Kerem, B., Alon, N., Rozmahel, R., Grzelczak, Z., Zielenski, J., Lok, S., Plavsic, N., y Chou, J. L. (1989). Identification of the cystic fibrosis gene: cloning and characterization of complementary DNA. *Science*, **245**(4922): 1066–1073.

Rommens, J. M., lannuzzi, M. C., Kerem, B., Drumm, M. L., Melmer, G., Dean, M., Rozmahel, R., Cole, J. L., Kennedy, D., Hidaka, N., Zsiga, M., Buchwald, M., Riordan, J. R., Tsui, L.-C., y Collins, F. S. (1989). Identification of the cystic fibrosis gene: chromosome walking and jumping. *Science*, **245**(4922): 1059–1065.

Rothemund, P. W. K. (1996). A DNA and restriction enzyme implementation of Turing machines. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, páginas 75–119.

Rouillard, J.-M., Zuker, M., y Gulari, E. (2003). Oligoarray 2.0: Design of oligonucleotides probes for DNA microarrays using a thermodynamics approach. *Nucleic Acids Research*, **31**(12): 3057–3062.

Roweis, S., Winfree, E., Burgoyne, R., Chelyapov, N. V., Goodman, M. F., Rothemund, P. W. K., y Adleman, L. M. (1998). A sticker-based model for DNA computation. *Journal of Computational Biology*, **5**(4): 615–629.

SantaLucia, J. J. y Hicks, D. (2004). The thermodinamics of DNA structural motifs. *Annual Review of Biophysics and Biomolecular Structure*, **33**: 415–440.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the 1st International Conference on Genetic Algorithms*, páginas 93–100, Hillsdale, NJ. ISBN: 0-8058-0426-9.

Schott, J. R. (1995). *Fault Tolerant Design Using Single and Multicriteria Genetic*

*Algorithm Optimization*. Tesis de doctorado, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Sugimoto, N., ichi Nakano, S., Katoh, M., Matsumura, A., Nakamuta, H., Ohmichi, T., Yoneyama, M., y Sasaki, M. (1995). Thermodynamic parameters to predict stability of RNA/DNA hybrid duplexes. *Biochemistry*, **34**(35): 11211–11216.

Talbi, E. (2009). *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley. ISBN: 9780470496909.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Tesis de doctorado, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Van Veldhuizen, D. A. y Lamont, G. B. (1999). On measuring multiobjective evolutionary algorithm performance. *Computer Engineering*, **1**: 204–211.

Vega-Briceño, L. E. (2004). CFTR: 15 años después del descubrimiento de un gen. *Rev Med Hered*, **15**(3): 159–165.

Wang, X. y Seed, B. (2003). Selection of oligonucleotide probes for protein coding sequences. *Bioinformatics*, **19**(7): 796–802.

Watkins, N. E., Kennelly, W. J., Tsay, M. J., Tuin, A., Swenson, L., Lee, H.-R., Morosyuk, S., Hicks, D. A., y SantaLucia, J. J. (2011). Thermodynamic contributions of single internal rA•dA, rC•dC, rG•dG and rU•dT mismatches in RNA/DNA duplexes. *Nucleic Acids Research*, **39**: 1894–1902.

Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC.

Xia, T., SantaLucia, J. J., Burkard, M. E., Kierzek, R., Schroeder, S. J., Jiao, X., Cox, C., y Turner, D. H. (1998). Thermodynamic parameters for an expanded nearest-

neighbor model for formation of RNA duplexes with watson-crick base pairs. *Biochemistry*, **37**(42): 14719–14735.

Yurke, B. y Mills, A. (2003). Using DNA to power nanostructures. *Genetic Programming and Evolvable Machines*, **4**: 111–122.

Zadeh, J. N., Steenberg, C. D., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., y Pierce, N. A. (2011). NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, **32**(1): 170–173.

Zhang, L., Volinia, S., Bonome, T., Calin, G. A., Greshock, J., Yang, N., Liu, C.-G., Giannakakis, A., Alexiou, P., Hasegawa, K., Johnstone, C. N., Megraw, M. S., Adams, S., Lassus, H., Huang, J., Kaur, S., Liang, S., Sethupathy, P., Leminen, A., Simossis, V. A., Sandaltzopoulos, R., Naomoto, Y., Katsaros, D., Gimotty, P. A., DeMichele, A., Huang, Q., Bützow, R., Rustgi, A. K., Weber, B. L., Birrer, M. J., Hatzigeorgiou, A. G., Croce, C. M., y Coukos, G. (2008). Genomic and epigenetic alterations deregulate microRNA expression in human epithelial ovarian cancer. *Proceedings of the National Academy of Sciences*, **105**(19): 7004–7009.

Zitzler, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Tesis de doctorado, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Zitzler, E. y Künzli, S. (2004). Indicator-based selection in multiobjective search. *8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, páginas 832–842. Springer.

Zitzler, E. y Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, **3**(4): 257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., y da Fonseca, V. (2003). Performance

assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, **7**(2): 117 – 132.