Artículo **Revista de Tecnología e Innovación**
Junio2016 Vol.3 No.7 38-45

Diseño de Circuitos Computacionales en Células Vivas usando Biología Sintética

MONDRAGÓN-LOZANO, Francisco*†, RAMÍREZ-CAMACHO, Jessica, VÁZQUEZ-HUERTA, María Elena y GONZALEZ-GUTIERREZ, Fidel.

Recibido Abril 18, 2016; Aceptado Junio 23, 2016

Resumen

Este trabajo se enfoca en el diseño de circuitos codificados de ADN y los métodos computacionales que contribuyen a su correcta funcionalidad, como lo son el algoritmo de Monte Carlo y el algoritmo de búsqueda por anchura (breadth-first). El diseño de los circuitos se implementó con un framework desarrollado por el Massachusetts Institute of Technology (MIT), específicamente para la creación de circuitos codificados de ADN, llamado *Cello* (abreviación para Cellular Logic). La optimización del circuito y la reducción del margen de error en la construcción de éstos, conlleva la búsqueda de la mejor opción y los algoritmos mencionados con anterioridad son utilizados por *Cello* para ese propósito. Al final, se logra un acercamiento al lenguaje *Cello*, con la implementación de circuitos básicos para conocer los alcances del framework y la revisión de los métodos computacionales que incrementan las capacidades de esta herramienta de biología sintética.

Circuitos Codificados de ADN, Biología Sintética, Programación Celular

Abstract

This research focuses on the design of DNA encoded circuits and the computational methods which contribute to its functionality, like Monte Carlo algorithm and breadth-first search algorithm. The design of the circuits was implemented using a framework developed by the Massachusetts Institute of Technology (MIT) specifically for the creation of those DNA encoded circuits, called *Cello* (stands for Cellular Logic). The optimization of the circuit and a reduction in the margin of error in building these, lead the search for the best option and the algorithms above are used by *Cello* for this reason. At the end, an approach to the Cello language was achieve, with the implementation of basic circuits to know the scope of the framework and a review of the computational methods that enhance the capacities of this synthetic biology tool.

DNA Encoded Circuits, Synthetic Biology, Cellular Programming

Citación: MONDRAGÓN-LOZANO, Francisco, RAMÍREZ-CAMACHO, Jessica, VÁZQUEZ-HUERTA, María Elena y GONZALEZ-GUTIERREZ, Fidel. Diseño de Circuitos Computacionales en Células Vivas usando Biología Sintética. Revista de Tecnología e Innovación 2016, 3-7: 38-45

*Correspondencia al Autor (Correo electrónico: 013012832@upq.edu.mx)
†Investigador contribuyendo como primer autor.

Introducción

El ADN es la huella única de cada organismo viviente, en él se encuentran todas las características que diferencian a ser vivo de otros. En la actualidad, es posible manipular las cadenas genéticas del ADN mediante técnicas de ingeniería, computación, biología, nanotecnología entre otras disciplinas. Dichas técnicas son estudiadas por una ciencia conocida como biología sintética, la cual busca mejorar el diseño y/o la implementación de sistemas biológicos que existen o no en la naturaleza.

El objetivo de la biología sintética puede ser alcanzado si se logran cambios en el funcionamiento celular. Quiere decir que una célula, al modificar su código genético, produciría nuevas reacciones ante diferentes estímulos ambientales, químicos, etc. Los circuitos de ADN son un medio para generar transformaciones importantes en la genética de una célula; éstos no son más que operaciones lógicas y teóricamente se comportan como los circuitos electrónicos, recibiendo señales, guiándolas a través de compuertas y generando una respuesta. Pero el diseño de los circuitos es laborioso y lleva mucho tiempo, gracias a las investigaciones del MIT en este tema se desarrolló un framework llamado *Cello* que maneja un lenguaje de programación para células vivas y facilita la construcción de los circuitos de ADN.

Cello está basado en Verilog, un lenguaje de descripción de hardware que funciona como modelador de sistemas electrónicos. Ocupa más herramientas como el lenguaje *Eugene* que ayuda con las restricciones de las secuencias de ADN para el circuito diseñado.

Los algoritmos de búsqueda por anchura, Monte Carlo y ascenso de colina (hill climbing) son las herramientas de búsqueda computacional que permiten el diseño del mejor circuito tomando en cuenta los parámetros iniciales dados y ya vienen programados dentro de la lógica de alto nivel de *Cello*. Entender este nuevo lenguaje permite realizar el diseño de circuitos de ADN y proporcionar una interpretación de su posible funcionamiento al aplicarlo a una célula viva sin necesidad de hacer pruebas dentro de un laboratorio. También da pauta para contribuir con diseños de circuitos óptimos y generar o descubrir nuevas funciones en determinadas células.

Al ser un tema amplio y con grandes oportunidades de investigación, entenderlo puede resultar una tarea poco sencilla, el estado del arte del tópico facilitará tener una mejor perspectiva para abordar el campo de la biología sintética, a través de una visión sobre el avance que ha tenido a lo largo del tiempo en el diseño de circuitos genéticos. Por medio de la metodología y su desarrollo se verá el trayecto recorrido para alcanzar los resultados presentados y las posibles líneas de acción futuras con respecto al tema.

Estado del Arte

En 1998, Leonard M. Adleman demostró la computación de ADN resolviendo el problema del camino hamiltoniano usando la expresión genética. En su trabajo explica que con el acercamiento que tuvo a los trabajos de Watson y Crick planteó un problema típico en computación, resolver un camino hamiltoniano donde el grafo a evaluar tenía únicamente cuatro nodos que representaban las ciudades de Atlanta, Boston, Chicago y Detroit.

ISSN-2410-3993

La diferencia fue que a cada ciudad se le asignó una cadena genética al igual que a la transición entre cada nodo. Sus pruebas de laboratorio fueron satisfactorias, pues las moléculas que ocupó lograron codificar el camino deseado y al final determinó que la computación de ADN tenía un futuro muy prometedor.

En el 2003 Erick Winfree publica en The Bridge su revisión sobre un algoritmo bioquímico basado en autoensamblaje de cristales heterogéneos. Winfree explora la idea del autoensamblaje molecular basado en estructuras de ADN por medio de la teoría conocida como Tiling Theory la cual habla del acomodo perfecto de figuras geométricas básicas en un plano infinito. En su trabajo también menciona que el autoensamblaje puede ser usado para colocar correctamente componentes moleculares (compuertas lógicas como AND, OR y NOT) que van unidas químicamente a alguna teja/losa de ADN.

Investigadores del J. Craig Venter Institute lograron en 2010 la construcción de la primera célula bacteriana sintética autoreplicable llamada *Mycoplasma mycoides*. Este avance demostró que el genoma puede ser diseñado por medios artificiales, como lo es una computadora. Actualmente siguen los avances en computación de ADN; así como el diseño y programación de circuitos genéticos. El caso del proyecto del MIT liderado por Cristopher Voigt salió a finales de marzo del presente año, en donde el lenguaje de programación celular *Cello* mejora el diseño de circuitos de ADN.

Diseño de Circuitos en Cello

Cello (Cellular Logic) es un framework que describe un lenguaje de programación para diseñar circuitos computacionales en células vivas. Su entrada es una especificación lógica de alto nivel escrita en Verilog.

Ese código es transformado a una tabla de verdad para producir un diagrama con los tipos de compuertas genéticamente disponibles para implementarla. Las compuertas son asignadas de forma inteligente usando diversos algoritmos de búsqueda basados en la heurística más alta. Por último el lenguaje *Eugene* arroja una o varias secuencias genéticas para el circuito diseñado. (https://github.com/CIDARLAB/cello/blob/master/README.md).

Para iniciar con la implementación de algún diseño se ocupa el repositorio del siguiente link: https://github.com/CIDARLAB/cello u ocupar la aplicación en línea de *Cello*: http://cellocad.org/

La entrada en Verilog puede ser escrita de cuatro maneras distintas: por casos, por asignación, estructurada o una combinación de la anteriores. Los siguientes códigos muestran la implementación de una operación lógica. En la figura 1, la entrada Verilog es por caso donde se asigna una tabla de verdad AND. La inicialización en el módulo A se hace con una variable de tipo output y dos de tipo input. Quiere decir que tendremos dos genes de entrada y sólo esperaremos un gen de salida.

```
module A(output out1,  input in1, in2);
 always@(in1,in2)
  begin
   case({in1,in2})
    2'b00: {out1} = 1'b0;
    2'b01: {out1} = 1'b0;
    2'b10: {out1} = 1'b0;
    2'b11: {out1} = 1'b1;
   endcase
  end
endmodule
```

Figura 1 Entrada Verilog por caso donde se asigna una tabla de verdad AND.

ISSN-2410-3993

La figura 2 muestra el módulo B, donde la implementación se realiza por asignación, la variable wire actúa como conector para alcanzar la última operación OR y que dará paso a la salida.

```
module B(output out1, input a, b);
  wire w0;
  assign w0 = a & ~b;
  assign out1 = w0 | (a & c);
endmodule
```

Figura 2 Entrada Verilog definida por asignación.

Por último, el ejemplo de la figura 3, el módulo C está de forma estructurada, se definió de manera literal las operaciones que las entradas ejecutarán.

```
module C(output out1,  input in1, in2);
  wire w1,w2,w3;
  not (w1, in1);
  not (w2, in2);
  nor (out1, w2, w1);
endmodule
```

Figura 3 Entrada Verilog estructurada.

En los circuitos de ADN las entradas son promotores; es decir, regiones del ADN que dictan el punto en el cual la ARN polimerasa comienza a transcribir un gen. Estos son representados (como cualquier otro gen) por una cadena de caracteres de determinada longitud. Cada carácter representa las bases nitrogenadas adenina (A), guanina (G), timina (T) y citosina (C). Las salidas del circuito esperadas suelen ser reguladores que generan respuestas a través de estímulos del entorno.

Algoritmos de Búsqueda

La lógica detrás del framework *Cello* tiene programados algoritmos de búsqueda clásicos que aseguran un diseño de circuito con las mejores características para que arroje la salida esperada. Este trabajo abarcará dos, el algoritmo de búsqueda por anchura (breadth-first) y el algoritmo de Monte Carlo ocupado en conjunto con un algoritmo de recocido simulado (simulated annealing). Los algoritmos de búsqueda trabajan por medio de la heurística más alta, esto quiere decir que por cada rama de decisión que se generé en la estructura de datos conocida como árbol, se le irá asignando una ponderación y seleccionan el siguiente paso tomando el mayor puntaje. A diferencia de Monte Carlo que es un método estadístico de optimización.

En *Cello*, el algoritmo de breadth-first se ejecuta de la siguiente manera: La búsqueda inicia la evaluación por las compuertas más próximas a las entradas y asigna ponderaciones una por una, donde la señal dispareja es rechazada. Cada vez que se ha aceptado una asignación del circuito, es decir que el camino que tomará el algoritmo es el más adecuado, vuelve a realizar el proceso. Así hasta que todas las compuertas han sido visitadas y la suma de las heurísticas más altas es elegida, se dibujan las compuertas y los resultados que corresponden a esa selección.

Simulated Annealing inicia con una asignación aleatoria, se seleccionando una compuerta al azar, las siguientes selecciones de compuertas son de manera también aleatoria. Monte Carlo revisa las probabilidades de todas las iteraciones hechas para determinar si alguna de las compuertas rechazadas puede llegar a ser una opción óptima y la toma en cuenta.

ISSN-2410-3993

Después de diseñar un circuito que permita emular el comportamiento celular, el siguiente paso es crear una cadena de ADN, aunque ensamblar cadenas de ADN largas puede resultar un proceso complejo de realizar al momento de implementarlas, pero para ello existen métodos que pueden simplificar este proceso como Golden-Gate, esto ayuda a que el proceso de ensamblaje tenga menos probabilidad de fallas, cabe mencionar que el récord de ensamblado fue gracias a la recombinación homológica en la que las secuencias de nucleótidos se intercambian entre dos moléculas similares o idénticas de ADN y así evita que la cadena de ADN sea larga y por ende, las probabilidades de fallas al momento de la implementación serán menores.

Resultados

Cello arroja los resultados del circuito procesado de diversas formas. Primero, se genera un UCF (por sus siglas en inglés, User Constrain File) que son los elementos que puede procesar el lenguaje hasta ese punto. Los genes disponibles para usar inmediatamente en la aplicación (tablas 1 y 2) pertenecen a cepas de *Escherichia Coli*, un organismo que cuenta con 880,000 pares de bases de ADN.

Gen	Secuencia ADN
pTac	AACGATCGTTGGCTGTGTTGACAATTAATCATCGGC TCGTATAATGTGTGGAATTGTGAGCGCTCACAATT
pTet	TACTCCACCGTTGGCTTTTTTCCCTATCAGTGATAGA GATTGACATCCCTATCAGTGATAGAGATAATGAGCA C
SrpR	CTGAAGCGCTCAACGGGTGTGCTTCCCGTTCTGATG AGTCCGTGAGGACGAAAGCGCCTCTACAAATAATTT TGTTTAAGAGTCTATGGACTATGTTTTCACAGAGGA GGTACCAGGATGGCACGTAAAACCGCAGCAGAAGC AGAAGAAACCCGTCAGCGTATTATTGATGCAGCACT GGAAGTTTTTGTTGCACAGGGTGTTAGTGATGCAAC CCTGGATCAGATTGCACGTAAAGCCGGTGTTACCCG TGGTGCAGTTTATTGGCATTTTAATGGTAAACTGGA AGTTCTGCAGGCAGTTCTGGCAAGCCGTCAGCATCC GCTGGAACTGGATTTTACACCGGATCTGGGTATTGA ACGTAGCTGGGAAGCAGTTGTTGTTGCAATGCTGGA TGCAGTTCATAGTCCGCAGAGCAAACAGTTTAGCGA AATTCTGATTTATCAGGGTCTGGATGAAAGCGGTCT GATTCATAATCGTATGGTTCAGGCAAGCGATCGTTT TCTGCAGTATATTCATCAGGTTCTGCGTCATGCAGTT ACCCAGGGTGAACTGCCGATTAATCTGGATCTGCAG ACCAGCATTGGTGTTTTTAAAGGTCTGATTACCGGT CTGCTGTATGAAGGTCTGCGTAGCAAAGATCAGCAG GCACAGATTATCAAAGTTGCACTGGGTAGCTTTTGG GCACTGCTGCGTGAACCGCCTCGTTTTCTGCTGTGT GAAGAAGCACAGATTAAACAGGTGAAATCCTTCGA ATAATTCAGCCAAAAAACTTAAGACCGCCGGTCTTG TCCACTACCTTGCAGTAATGCGGTGGACAGGATCGG CGGTTTTCTTTTCTCTTCTCAATCTATGATTGGTCCA GATTCGTTACCAATTGACAGCTAGCTCAGTCCTAGG TATATACATACATGCTTGTTTGTTTGTAAAC

Tabla 1 Codificación del ADN de los genes usados para el primer circuito.

Se probaron diversos casos. El primero fue una entrada Verilog por caso, la tabla de verdad fue una AND y los genes de entrada fueron promotores *pTac* que es un híbrido de otros dos promotores de la *Eschericia Coli* mucho más eficiente y *pTet* que juega un papel importante en la activación del gen.

Para la salida, esperamos sea un regulador *SrpR*, funcionalidad que ayuda con la creación de una enzima que a su vez se involucra con la producción de una molécula llamada *tetrahydrobiopterin*.

La búsqueda del mejor circuito se completó en 94467 milisegundos y de las 50 trayectorias realizadas en total, tomando en cuenta todas las iteraciones, el mejor resultado de asignación fue de 328.2888.

ISSN-2410-3993

El circuito construido, junto con las heurísticas más altas por compuerta se aprecian en las imágenes 3 y 4. Gracias al diagrama Eugene que genera *Cello* (imágenes 1 y 4), gráficamente se identifica cómo interactúan los componentes dentro del circuito.

Figura 4 Representación Eugene del circuito generado. Muestra su expresión genética de forma gráfica.

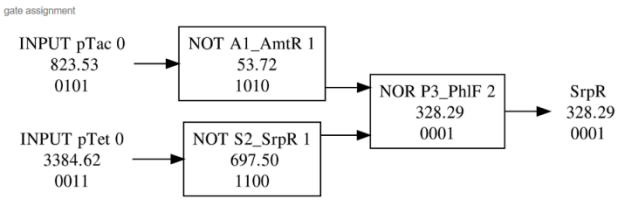

Figura 5 Asignación de compuertas. Observamos los valores que los algoritmos de búsqueda seleccionaron como los óptimos para cada compuerta lógica.

Para el circuito número dos se ocuparon los promotores *pLuxStar* y *pBAD*, esperando una respuesta de fluorescencia roja (*RFP*) como se ve en la tabla 2. El tiempo de ejecución del circuito fue de 94990 milisegundos y la asignación con el mejor puntaje se hizo de 184.3090 (imagen 3).

Gen	Secuencia ADN
pLuxStar	ATAGCTTCTTACCGGACCTGTAGGATCGT ACAGGTTTACGCAAGAAAATGGTTTGTT ACTTTCGAATAAA
pBAD	ACTTTTCATACTCCCGCCATTCAGAGAAG AAACCAATTGTCCATATTGCATCAGACA TTGCCGTCACTGCGTCTTTTACTGGCTCT TCTCGCTAACCAAACCGGTAACCCCGCTT ATTAAAAGCATTCTGTAACAAAGCGGGA CCAAAGCCATGACAAAAACGCGTAACAA AAGTGTCTATAATCACGGCAGAAAAGTC CACATTGATTATTTGCACGGCGTCACACT TTGCTATGCCATAGCATTTTTATCCATAA GATTAGCGGATCCTACCTGACGCTTTTTA TCGCAACTCTCTACTGTTTCTCCATACCC GTTTTTTTGGGCTAGC
RFP	CTGAAGTGGTCGTGATCTGAAACTCGAT CACCTGATGAGCTCAAGGCAGAGCGAAA CCACCTCTACAAATAATTTTGTTTAATAC TAGAGTCACACAGGAAAGTACTAGATGG CTTCCTCCGAAGACGTTATCAAAGAGTTC ATGCGTTTCAAAGTTCGTATGGAAGGTTC CGTTAACGGTCACGAGTTCGAAATCGAA GGTGAAGGTGAAGGTCGTCCGTACGAAG GTACCCAGACCGCTAAACTGAAAGTTAC CAAAGGTGGTCCGCTGCCGTTCGCTTGG GACATCCTGTCCCCGCAGTTCCAGTACG GTTCCAAAGCTTACGTTAAACACCCGGC TGACATCCCGGACTACCTGAAACTGTCCT TCCCGGAAGGTTTCAAATGGGAACGTGT TATGAACTTCGAAGACGGTGGTGTTGTT ACCGTTACCCAGGACTCCTCCCTGCAAG ACGGTGAGTTCATCTACAAAGTTAAACT GCGTGGTACCAACTTCCCGTCCGACGGT CCGGTTATGCAGAAAAAAACCATGGGTT GGGAAGCTTCCACCGAACGTATGTACCC GGAAGACGGTGCTCTGAAAGGTGAAATC AAAATGCGTCTGAAACTGAAAGACGGTG GTCACTACGACGCTGAAGTTAAAACCAC CTACATGGCTAAAAAACCGGTTCAGCTG CCGGGTGCTTACAAAACCGACATCAAAC TGGACATCACCTCCCACAACGAAGACTA CACCATCGTTGAACAGTACGAACGTGCT GAAGGTCGTCACTCCACCGGTGCTTAAT AACAGATAAAAAAAATCCTTAGCTTTCG CTAAGGATGATTTCT

Tabla 2 Codificación del ADN de los genes usados para el segundo circuito.

ISSN-2410-3993

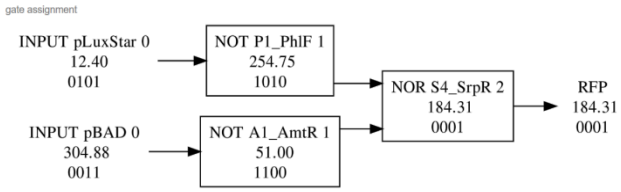

Figura 6 Asignación de compuertas. Observamos los valores que los algoritmos de búsqueda seleccionaron como los óptimos para cada compuerta lógica.

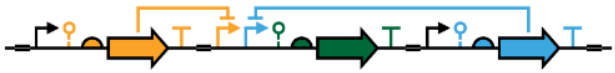

Figura 7 Representación Eugene del circuito generado. Muestra su expresión genética de forma gráfica.

Anexos

Promotor: Región del ADN que determina el punto en el que el ARN polimerasa comienza a transcribir un gen.

ARN Polimerasa: Enzima encargada de la transcripción del ADN.

Transcripción: Proceso en el que la información genética del ADN pasa por el ARN mensajero. Este es el primer paso para la síntesis de proteínas.

Reguladores: Secuencias contiguas a un gen que mantienen bajo control la tasa de transcripción del mismo.

Agradecimiento

Agradecemos el apoyo y financiamiento de la Universidad Politécnica de Querétaro.

Conclusiones

A través del desarrollo de las herramientas computacionales programadas como parte de la lógica de *Cello*, es posible diseñar circuitos de ADN con una reducción de tiempo significativa así como predecir su comportamiento.

El comportamiento del circuito puede llegar a ser lento por las muchas iteraciones que tiene que realizar para su desarrollo, los algoritmos de búsqueda como Monte Carlo y breathd-first impactan de manera significativa al momento de realizar este proceso de una manera más eficiente, optimizando recursos y tiempo de investigación. Estos algoritmos traen beneficios a la biología sintética ya que los algoritmos de búsqueda se encargan de hallar la mejor ruta y además de optimizar recursos puede dar como resultado una cadena de ADN más corta que esto a su vez trae beneficios al momento de implementarla en una célula viva. Por todo lo anterior basta decir que estos algoritmos de búsqueda actúan de manera positiva al momento de realizar las iteraciones en la búsqueda del circuito funcional. Las ciencias computacionales nos permiten realizar estos avances en la reingeniería de la vida, se pretende como trabajo futuro continuar con la investigación de estos circuitos de ADN con ayuda de estos algoritmos de búsqueda además de implementarlos de manera que su reacción hacia la célula de algún organismo sea benéfica cuando esta esté dañada, con todos los avances tecnológicos que crecen exponencialmente, esta implementación se puede ver de una perspectiva cada vez más cercana a la realidad, obteniendo resultados que se puedan adaptar a las necesidades de la humanidad y darle un cambio a la vida.

Referencias

Adleman, Leonard M. "Computing With DNA". Scientific American (1998).

Winfree, Erick. “DNA Computing by Self-Assembly”. The Bridge (2003).

Nielsen Alee, Der Bryan, Shin Jonghyeon, Vaidyanathan Prashant, Paranalov Banya, Strychalski Elizabeth, Ross David, Densmore Douglas, Voigt Cristopher. “Genetic Circuit Design Automation”. Science (2016).

ISSN-2410-3993

Russell, Stuart. “Artificial Intelligence: A Modern Approach”. 3rd Edition. (p. 75-75, 111-114).

"Cello". CIDAR at Boston University. N.p., 2016. Web. 1 Sept. 2016.

ISSN-2410-3993