



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

DISEÑO E IMPLEMENTACIÓN DE FUNCIONES BIOLÓGICAS ELEMENTALES EN COLONIAS BACTERIANAS PARA LA INGENIERÍA DE MORFOGÉNESIS

ISAAC NATÁN NÚÑEZ QUIJADA

Tesis para optar al grado de
Magister en Ciencias de la Ingeniería

Profesor Supervisor:
FERNÁN FEDERICI NOE
TIMOTHY JAMES RUDGE

Santiago de Chile, Junio, 2017

© 2017, Isaac Natán Núñez Quijada



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

DISEÑO E IMPLEMENTACIÓN DE FUNCIONES BIOLÓGICAS ELEMENTALES EN COLONIAS BACTERIANAS PARA LA INGENIERÍA DE MORFOGÉNESIS

ISAAC NATÁN NÚÑEZ QUIJADA

Tesis presentada a la Comisión integrada por los profesores:

FERNÁN FEDERICI NOE

TIMOTHY JAMES RUDGE

DANIEL GARRIDO CORTES

JUAN KEYMER VERGARA

CRISTIAN ESCAURIAZA MESA

Para completar las exigencias del grado de
Magister en Ciencias de la Ingeniería

Santiago de Chile, Junio, 2017

A los que viven la ciencia

AGRADECIMIENTOS

En primer lugar, agradezco a todos los que participaron en el desarrollo de esta tesis. Particularmente a Ilenne del Valle (Pontificia Universidad Católica de Chile) por el trabajo realizado en la caracterización de la técnica de segregación (Figuras 4-6 y 4-7), a Anton Kan (University of Cambridge) por llevar a cabo la caracterización de la inducción de los plásmidos SEG (Figura 4-5), y a Atri Choksi (Stanford University) por elaborar la segregación de cuatro plásmidos (Figura 4-8). Muchas gracias por la consecuente colaboración en el artículo publicado (Anexo A).

Agradecer especialmente a Tamara Matute por el desarrollo de toda la tecnología CRISPRi involucrada en este trabajo, clave para alcanzar el objetivo común en nuestras tesis y la publicación del artículo científico.

A mi tutor Fernan Federici, por su dedicación, apertura y confianza, por su guía, por enseñarme a hacer ciencia y a ser un científico para la sociedad.

A mi tutor Timothy Rudge, por su guía, apoyo, preguntas y enseñanzas. Especialmente, por su disponibilidad y dirección en el análisis de imágenes y modelamiento.

Al profesor Ricardo Pérez por todo el respaldo y apoyo en el iniciar este magister.

Al profesor Rodrigo Gutierrez por creer en nosotros desde un inicio y otorgarnos todo su apoyo en nuestro desarrollo científico e investigación.

A Bernardo Pollak por maravillarme con la investigación y enseñarme el trabajo experimental, sin él nada de esto habría comenzado.

Agradecer a mi equipo y experiencia iGEM, por ser parte fundamental en mi desarrollo.

A Tamara por ser mi compañera en todo, por su apoyo, rigurosidad científica y dedicación.

Finalmente, me gustaría agradecer a mi familia por su comprensión y apoyo en todo el transcurso de mi formación, por su interés en mi investigación, y por enseñarme a hacer las cosas con todo el corazón y maestría.

Esta tesis fue desarrollada con el apoyo financiero de: CONICYT-PAI/Concurso Nacional de Apoyo al Retorno de Investigadores/as desde el Extranjero Folio 82130027 y Fondecyt Iniciación 11140776

INDICE GENERAL

Pág.

DEDICATORIA	ii
AGRADECIMIENTOS	iii
INDICE DE TABLAS	vi
INDICE DE FIGURAS.....	vii
RESUMEN.....	ix
ABSTRACT.....	x
1. INTRODUCCIÓN.....	1
1.1 Principios de biología sintética e ingeniería morfogénica	1
1.1.1 Biología sintética	1
1.1.2 Morfogénesis	3
1.1.3 Ingeniería morfogénica	4
1.2 Bases para la formación de funciones genéticas modulares	5
1.2.1 Instrucción de funciones genéticas	5
1.2.2 Nomenclatura y unidades de transcripción.....	7
1.2.3 Funciones genéticas modulares ortogonales.....	8
1.3 Simulación de sistemas multicelulares.....	9
1.4 Sistema CRISPRi	12
2 HIPOTESIS Y OBJETIVOS	14
2.1 Hipótesis.....	14
2.2 Objetivo General	14
2.3 Objetivos específicos	14
3 METODOLOGIA.....	15
3.1 Ensamblaje y preparación de ADN.....	15
3.2 <i>Colony</i> PCR.....	16
3.3 Condiciones de crecimiento	16

3.4	Transformación y Co-transformación bacteriana.....	19
3.5	Registro de imágenes	21
3.6	Análisis de imágenes en ensayo de cambios morfológicos	22
3.7	Ensayos de cepas auxótrofas en metionina e inducción de cambios morfológicos	23
3.8	Implementación de módulos en CellModeller	24
3.9	Caracterización de la segregación de plásmidos SEG#	25
4	RESULTADOS	28
4.1	Abstracción y desacoplamiento.....	28
4.2	Ruptura de simetría y capa formadora de patrones	29
4.2.1	Diseño de la capa formadora de patrones	30
4.2.2	Implementación en modelo biofísico.....	32
4.2.3	Establecimiento de dominios celulares mediante ruptura de simetría.....	33
4.3	Capa actuadora y control de expresión genética dominio específica.....	39
4.3.1	Ingeniería de la capa actuadora.....	39
4.3.2	Complementación metabólica dominio específica	41
4.3.3	Control de expresión genética dominio específica con RNAPT7 .	43
4.4	Prototipado de regulación morfogenética	48
5	CONCLUSIONES Y PERSPECTIVAS	55
	BIBLIOGRAFIA	58
	A N E X O S	69
	Anexo A: Artículo ISI.....	70
	Anexo B: Rutinas de MATLAB.....	81
	Anexo C: Rutinas usadas en Cellmodeller.....	92
	Anexo D: Primers utilizados	102

INDICE DE TABLAS

	Pág.
Tabla 1.1 : Listado de algunos plásmidos comúnmente usados en <i>E. coli</i>	7
Tabla 3.1 : Receta medio LB.....	17
Tabla 3.2: Receta Medio M9.....	17
Tabla 3.3 : Soluciones stock.....	18
Tabla 3.4: Filtros de fluorescencia utilizados para cada proteína fluorescente.....	21
Tabla 3.5: Especificaciones de las longitudes de onda utilizadas en la segregación de cuatro plásmidos.....	22
Tabla 4.1: Probabilidad de obtener colonias bipartitas a partir de diferente número de copias de plásmidos en célula fundadora.	33

INDICE DE FIGURAS

	Pág.
Figura 1-1: Topología y simbología típica de unidades transcripcionales de ADN.....	8
Figura 1-2: Simulación en CellModeller de colonia en crecimiento	10
Figura 1-3: Esquema del algoritmo central de CellModeller.....	11
Figura 3-1: Modulo <i>update</i> implementado en simulaciones.....	25
Figura 4-1: Desacoplamiento de mecanismos morfogénéticos en funciones elementales.....	29
Figura 4-2: Mecanismo de segregación celular	31
Figura 4-3: Simulación de colonia bacteriana en crecimiento	31
Figura 4-4: Colonias bacterianas multisector para 4 a 10 copias por célula	32
Figura 4-5: Plásmido desarrollado con número de copias inducible por arabinosa.....	34
Figura 4-6: Fases y optimización de protocolo segregación	36
Figura 4-7: Falsos positivos.	37
Figura 4-8: Serie de vectores “SEG” diseñados y definición de hasta cuatro sectores dentro de una colonia bacteriana.....	38
Figura 4-9: Esquema del plásmido pAct.....	40
Figura 4-10: Implementación de la capa actuadora	40
Figura 4-11: Expresión parcial o total de la ruta bio-sintética de violaceína.....	42
Figura 4-12: Complementación metabólica dominio específica.....	43
Figura 4-13: Esquema unidades transcripcionales incluidas en S8T7 y pAct3	44
Figura 4-14: Control de la expresión de mCherry con RNAPT7 e IPTG	45
Figura 4-15: Ensayo de regulación genética dominio específica con RNAPT7	46

Figura 4-16: Regulación dominio específica y temporal utilizando RNAPT7 dividida.....	47
Figura 4-17: Regulación dominio específica de hasta dos genes mediante RNAPT7	48
Figura 4-18: Esquema del sistema de inducción de cambios morfogénéticos.....	49
Figura 4-19: Inducción de cambios morfogénéticos de manera dominio específica en colonias bacterianas	51
Figura 4-20: Resultados representativos de cambios morfogénéticos. Se presentan imágenes de las morfologías más sobre-representadas, tanto en simulaciones como en data experimental, en tres experimentos independientes. Barra = 500 μ m.....	52
Figura 4-21: Serie de tiempo de efecto morfogénético	54
Figura A- 1: Ruptura de simetría a partir de célula fundadora con dos plásmidos distintos en su interior.	93
Figura A- 2: Ruptura de simetría a partir de célula fundadora con tres plásmidos distintos en su interior.	95
Figura A- 3: Ruptura de simetría a partir de célula fundadora con cuatro plásmidos distintos en su interior.	97
Figura A- 4: Ruptura de simetría y regulación del crecimiento celular de manera dominio específica desde el inicio.	99
Figura A- 5: Ruptura de simetría y regulación del crecimiento celular de manera dominio específica con control temporal.	101

RESUMEN

La ingeniería biológica se constituye sobre la dicotomía entre el enfoque estructurado y determinista de la ingeniería, y la naturaleza autoorganizada y estocástica de los sistemas biológicos que utiliza. La ingeniería morfogenética se define en medio de este conflicto, aprovechando el potencial de la autoorganización para formar sistemas altamente estructurados, mediante la manipulación de las unidades que los componen y definen. Por otra parte, la biología sintética pretende crear herramientas y formalismos para mejorar la trazabilidad y capacidad de diseño del sustrato biológico. En esta tesis se exploran estas dos áreas para crear una plataforma de prototipado de funciones morfogenéticas, que permita tanto estudiar sus principios como diseñar estructuras de nivel superior a partir de la programación de una célula fundadora. Para este fin, se propone desacoplar la morfogénesis en funciones elementales modulares, como lo son la ruptura de simetría y la regulación celular dominio específica. Concretamente, se desarrolló un sistema genético en dos capas: una formadora de patrones, fundamentada en la segregación de instrucciones celulares, y otra capaz de modificar el comportamiento celular en base a la interacción con los componentes co-localizados de la primera capa. Estas funciones fueron incluidas como módulos computacionales en CellModeller, lo que permitió explorar *in silico* su funcionamiento y guiar el diseño. Este enfoque de desacoplamiento y exploración computacional, unido a la inclusión de un sistema modular y combinatorial de ensamblaje de ADN, facilitaron el ciclo de “diseño, fabricación y testeo” de funciones morfogenéticas. Para poder aplicar estas herramientas, en primer lugar, se testeó y estableció su funcionamiento de manera independiente. Luego, se prototiparon funciones simples como “expresión genética dominio específica” mediante RNAPT7 y complementación metabólica. Finalmente, como prueba de concepto, se utilizó un sistema CRISPRi capaz de regular la tasa de crecimiento celular para modificar la morfología de las colonias bacterianas.

Palabras Claves: Biología sintética, ingeniería morfogenética, instrucciones modulares de ADN, colonias bacterianas, ruptura de simetría.

ABSTRACT

Engineering biological substrates poses a challenge between the determinism of engineering approaches and the self-organized and stochastic nature of biological systems. Morphogenetic engineering sits at this intersection, taking advantages of the potential of self-organization to form highly structured systems through the manipulation of basic units and building blocks. On the other hand, synthetic biology seeks to create tools and formalisms to improve the tractability and capabilities to design biological substrates. In this thesis, these two areas were explored to create a platform for prototyping morphogenetic functions to study and engineer high level structures from individual founding cells. To this end, it was proposed to decouple morphogenesis into elementary functions such as symmetry-breaking and domain-specific regulation of cellular processes. Concretely, a genetic system composed of two layers was created: one pattern-forming, by segregating cellular instructions, and a second capable of modifying cellular behavior by interactions of components present in the two layers.

These functions were successfully abstracted into computational modules to be run in CellModeller, which allowed the *in silico* exploration of performance and the guidance of the design process. This approach based on decoupling and computational guidance, together with the use of a modular and combinatorial DNA fabrication system, facilitated the design-build-test cycle. To be able to apply these tools, it was necessary to test their independent performance. Next, simple functions such as RNAPT7 gene regulation and metabolic complementation were prototyped. Finally, a CRISPRi system was used as a proof of concept of tractable morphogenetic changes based on the regulation of cell growth.

Keywords: Synthetic biology, morphogenetic engineering, modular DNA instructions, bacterial colonies, symmetry breaking.

1. INTRODUCCIÓN

1.1 Principios de biología sintética e ingeniería morfogénica

1.1.1 Biología sintética

La biología sintética propone la aplicación de los principios de ingeniería - modularidad, abstracción y desacoplamiento - para la manipulación y diseño de nuevas funciones en sistemas biológicos (Endy, 2005). Es decir, pretende el diseño racional y predecible del sustrato biológico, mediante la caracterización y utilización de partes genéticas modulares que permitan ensamblar sistemas más complejos con una función deseada.

Este enfoque ha revolucionado el campo de la bioingeniería en los últimos años y ha tenido éxito en el establecimiento de formalismos que aumentan la modularidad, trazabilidad y capacidad de diseño de sistemas biológicos (Del Vecchio, 2015; Jack et al., 2015; Lou, Stanton, Chen, Munsky, & Voigt, 2012; A. K. Nielsen et al., 2016). Con ello se está cambiando radicalmente el pensamiento respecto a la materia viva, pasando de ser una mera fuente de materias primas a ser considerada un sustrato programable, capaz de realizar computación, fabricación de materiales responsivos y síntesis de compuestos químicos complejos (Benenson, 2012; Chen et al., 2014; Galanie, Thodey, Trenchard, Interrante, & Smolke, 2015; Siuti, Yazbek, & Lu, 2014).

Más aún, la biología sintética no solo ha permitido el diseño de sistemas biológicos, sino que ha cambiado la manera en la que podemos interrogar y entender su funcionamiento natural, a través del diseño de redes sintéticas genéticas y/o proteicas con fines de investigación científica (Collins, Gardner, & Cantor, 2000; M. Elowitz & Lim, 2010; Mukherji & van Oudenaarden, 2009). Esto

permite no solo estudiar la biología “como es”, sino que interrogar sus límites y entender cómo podría “llegar a ser”.

No obstante, la biología sintética es principalmente una disciplina de ingeniería, cuyo foco es crear sistemas nuevos más que trabajar en la descripción de los sistemas naturales (Church, Elowitz, Smolke, Voigt, & Weiss, 2014). Para ello, se centra en la caracterización de las partes de manera que su descripción sea útil para predecir y evaluar su desempeño en los sistemas diseñados, con el fin de poder escalar la complejidad de diseño de funciones nuevas (Arkin, 2013). Existen diversos ejemplos del desarrollo de este enfoque, desde el *toggle switch* (Gardner, Cantor, & Collins, 2000) y los osciladores (M. B. Elowitz & Leibler, 2000; Stricker et al., 2008), hasta sistemas capaces de integrar múltiples señales (Xie, Wroblewska, Prochazka, Weiss, & Benenson, 2011) o la posibilidad de construir circuitos genéticos complejos a partir de un diseño totalmente *in silico* (Nielsen et al., 2016).

Para conseguir estos objetivos ha sido necesario desarrollar novedosas técnicas de fabricación de ADN, (Gibson et al., 2009; Kosuri & Church, 2014; Weber, Engler, Gruetzner, Werner, & Marillonnet, 2011) herramientas de diseño asistido por computadoras y especificaciones para la elaboración de redes genéticas (Brophy & Voigt, 2014). Sin embargo, la escalabilidad de dichas redes a nivel celular se ha visto limitada por efectos de carga metabólica o interacciones indeseadas entre los elementos insertados (Purnick & Weiss, 2009). Por ello, los diseños distribuidos a nivel multicelular surgen como una solución a este problema de escalabilidad.

Aún quedan muchos aspectos que abordar en torno al funcionamiento efectivo de los circuitos genéticos, y probablemente uno de los más desafiantes sea cambiar el paradigma de funcionamiento en organismos individuales a circuitos que funcionen en forma verdaderamente multicelular (Church et al., 2014). El éxito de

esta perspectiva podría tener aplicaciones en la ingeniería de tejidos, *biofilms*, consorcios bacterianos y biomateriales, ampliando el horizonte de lo diseñable.

1.1.2 Morfogénesis

Como su nombre sugiere, la morfogénesis corresponde al proceso de generación autoorganizada de estructuras y formas en los seres vivos, que se fundamenta en la ruptura de simetría, la formación de patrones y la diferenciación celular. Este proceso se encuentra asociado a diversos mecanismos, tanto químicos como físicos, que permiten el establecimiento de información espacial y organización de las células en el sistema (Goodwin, Kauffman, & Murray, 1993; Howard, Grill, & Bois, 2011). Entre los mecanismos podemos encontrar: interacciones de superficie como fuerzas impulsoras de clasificación y ordenamiento celular (Steinberg, 1970); densidad y motilidad (Liu et al., 2011); y la distribución de morfógenos (factores químicos o macromoléculas), como determinantes del estado, crecimiento y diferenciación celular (Wolpert, 1969).

Este último, ha sido el concepto central en la formación de patrones desde que Alan Turing (1952) propusiera un sistema de reacción y difusión como mecanismo para el establecimiento de distribuciones espaciales de morfógenos a partir de sistemas inicialmente cuasi-homogéneos. Este sistema y sus secuelas han sido ampliamente estudiados (Gierer & Meinhardt, 1972), así como los mecanismos que permiten interpretar dicho gradiente morfogénico (Saka & Smith, 2007; Schaerli, Munteanu, Cotterell, Sharpe, & Isalan, 2014). Basándose en esta lógica, ha sido posible establecer patrones espaciales artificiales en colonias bacterianas mediante el uso de *quorum sensing* (Basu, Gerchman, Collins, Arnold, & Weiss, 2005), pero solo recientemente se han prototipado mecanismos verdaderamente auto-organizados que se fundamentan en el uso del morfógeno como señal temporal (Payne et al., 2013), o basados en la integración de densidad y motilidad

(Liu et al., 2011), faltando mucho por explorar en torno al diseño, interacción e implementación de patrones.

1.1.3 Ingeniería morfogenética

Los sistemas biológicos son de gran interés como fuente de inspiración y objeto de diseño. Como fuente de inspiración, la comprensión del funcionamiento de dichos sistemas ha permitido aplicar sus principios para el diseño de máquinas, lógicas organizacionales y algoritmos computacionales (Adamatzky & Jones, 2010; Fister Jr, Yang, Fister, Brest, & Fister, 2013; Olmos & Kingdom, 2004; Tero et al., 2010; Zeng, Li, Chen, Yang, & Li, 2010). Como objeto de diseño, las capacidades son más modestas, pues la manipulación racional y predictiva de seres vivos nunca está exenta de la complejidad de su funcionamiento y comportamiento autónomo. En gran parte, dicha limitación surge de la dicotomía que existe entre la naturaleza auto-organizada y estocástica de los seres vivos, y la racionalidad determinística deseada en los diseños estructurados de ingeniería (Doursat, Sayama, & Michel, 2013).

La ingeniería morfogenética es un campo emergente en las ciencias de la vida que está situado en el centro de esta dicotomía. Esta busca el diseño, planificación e implementación de patrones, funciones y estructuras autoorganizadas en sistemas compuestos por múltiples unidades de comportamiento estocástico como lo son los tejidos celulares o las colonias bacterianas. En último término, intenta aprovechar el potencial de la auto-organización para formar sistemas altamente estructurados, en base a la manipulación de las propiedades de las unidades que los componen y definen (Doursat, 2012).

Sin embargo, la morfogénesis involucra procesos multi-paralelos, colectivos y emergentes en diversas escalas espaciales y temporales (Goodwin, B. C.,

Kauffman, S., & Murray, 1993). Para afrontar estos desafíos se hacen necesarias nuevas herramientas y enfoques que permitan abordar el problema de diseñar mecanismos morfogenéticos en un modo trazable, predecible y susceptible a ciclos de diseño-fabricación-testeo. De este modo, aplicar principios de ingeniería como abstracción y elaboración de funciones elementales modulares a través de la biología sintética, se presenta como una oportunidad para abordar el problema y desarrollar un sistema de prototipado de mecanismos morfogenéticos.

1.2 Bases para la formación de funciones genéticas modulares

Sin lugar a duda una de las piezas clave en el desarrollo de la biología sintética ha sido la capacidad de leer, escribir y ensamblar fragmentos de ADN. Estos avances han cambiado la manera en la que se diseñan los sistemas biológicos (Church et al., 2014). Específicamente, la síntesis de ADN y las metodologías de ensamblaje han incrementado la capacidad, escala y complejidad de las instrucciones genéticas, permitiendo fabricar secuencias artificiales de gran tamaño y alta fidelidad (Kosuri & Church, 2014).

1.2.1 Instrucción de funciones genéticas

Típicamente, la forma de instruir funciones en microorganismos es a través de la inserción de secuencias genéticas en vectores de expresión extra-cromosomales llamados plásmidos (Schumann, 2008). Estos se caracterizan por tener al menos dos propiedades: un origen de replicación (Ori) que sea compatible con la maquinaria celular y permita su multiplicación, y una ventaja selectiva que asegure la perdurabilidad de éste en la célula en un ambiente determinado (normalmente un gen que codifique una resistencia a antibióticos) (Thomas & Summers, 2008). La construcción y uso de colecciones de plásmidos permite introducir funciones de

manera modular en las células, sin necesidad de modificar directamente el genoma del microorganismo.

El origen de replicación utilizado en cada plásmido es de vital importancia, pues no solo cumple la función mencionada, sino que además se relaciona con el número de copias que existirán de él dentro del microorganismo. Consecuentemente, esto determina el número de copias de los genes que se codifican en dicho plásmido, lo que influye en sus niveles de expresión (Schendel, Baude, & Flickinger, 1989). Este número de copias es determinado por el *replicón*, que es el conjunto de la secuencia nucleotídica del Ori más los elementos involucrados en su regulación (Del Solar, Giraldo, Ruiz-Echevarría, M. J., Espinosa, & Díaz-Orejas, 1998). Si dos plásmidos insertados tienen la misma maquinaria de replicación, serán indistinguibles para la célula en este aspecto, y compartirán el número total de copias. Esto genera una distribución y herencia asimétrica y/o pérdida de uno de ellos a lo largo de las divisiones celulares (Velappan, Sblattero, Chasteen, Pavlik, & Bradbury, 2007).

Debido a lo anterior, los plásmidos se clasifican en grupos de incompatibilidad determinados por su replicón (Thomas & Summers, 2008). En la **Tabla 1.1** se presenta una lista con algunos de los vectores de uso más cotidiano, cuyo número de copia característico puede variar desde valores cercanos a uno, hasta varios cientos. Un número de copias alto significa un gran gasto de recursos para la célula (alta carga metabólica) (Palomares & Estrada-Mondaca, S. Ramirez, 2004), por lo que en la expresión de circuitos genéticos usualmente resultan de mayor interés aquellos replicones con un valor característico medio o pequeño.

Tabla 1.1 : Listado de algunos plásmidos comúnmente usados en *E. coli*

Vector Típico	N° de copias	ORI	Grupo de incompatibilidad*
pUC	~500-700	pMB1 (derivado)	A
pBR322	~15-20	pMB1	A
pET	~15-20	pBR322	A
pGEX	~15-20	pBR322	A
pColE1	~15-20	ColE1	A
pR6K	~15-20	R6K*	C
pACYC	~10	p15A	B
pSC101	~5	pSC101	C
pBluescript	~300-500	ColE1 (derivado) y F1	A
pGEM	~300-500	pUC y F1	A

Tabla modificada de Addgene (<http://blog.addgene.org/plasmid-101-origin-of-replication>).

ORI: Origen de replicación. * El nombre del grupo de compatibilidad es arbitrario.

1.2.2 Nomenclatura y unidades de transcripción

En cuanto a la estructura de las funciones genéticas insertadas, estas pueden ser abstraídas en unidades transcripcionales (TU, *Transcriptional Unit*), compuestas por: un promotor, un sitio de unión ribosomal (RBS, *Ribosome Binding Site*), región codificante de proteína (CDS: *Coding DNA Sequence*) y terminador transcripcional. La estructura y simbología utilizada en las TU se puede observar en la Figura 1-1. El promotor es una región de especial interés, pues es el sitio con afinidad por la unión de la enzima ARN polimerasa e inicio de la transcripción, que por lo demás, puede contener múltiples secuencias nucleotídicas que estén involucradas con la regulación de su expresión. Actualmente se cuenta con un registro de partes genéticas estándares (<http://parts.igem.org/>), donde es posible obtener la secuencia y características de cientos de cada uno de estos elementos moleculares.

T.U.	
Promotor	
RBS	
CDS	
Terminador	

Figura 1-1: Topología y simbología típica de unidades transcripcionales de ADN. TU: Unidad transcripcional; RBS: sitio de unión ribosomal; CDS: región codificante de proteína. Simbología de acuerdo al Synthetic Biology Open Language (SBOL, <http://sbolstandard.org/>)

1.2.3 Funciones genéticas modulares ortogonales

Un aspecto clave de la modularidad funcional es la capacidad de combinar los componentes sin generar interferencia o interacciones inesperadas entre ellos (Lucks, Qi, Whitaker, & Arkin, 2008). Dado que las funciones genéticas se fundamentan en relaciones físico-químicas complejas entre macromoléculas biológicas (*i.e.* proteína-proteína, proteína-ADN, ARN-ADN, etc.), en general es difícil predecir y evitar interacciones indeseadas entre elementos o con la maquinaria celular endógena de los organismos en el que serán insertados (Lu, Khalil, & Collins, 2009).

Superar esta dificultad ha sido uno de los principales ámbitos de desarrollo en la ingeniería de sistemas biológicos complejos (Clancy, K., & Voigt, 2010). Mediante la búsqueda de elementos en sistemas naturales (Bayer et al., 2009; Nielsen, Segall-Shapiro, & Voigt, 2013) y ensayos a gran escala de variaciones aleatorias en la estructura de los componentes (Rhodius et al., 2013), se han desarrollado diversas partes genéticas capaces de desempeñar funciones

específicas de manera ortogonal. Entre los componentes elaborados se pueden encontrar: reguladores genéticos proteicos que permiten integrar múltiples señales (Stanton et al., 2014; Wang, Kitney, Joly, & Buck, 2011), terminadores de transcripción de alta eficiencia, sistemas de señalización inter-celular (Grant et al., 2016) y sistemas de expresión genética paralelos a la maquinaria de expresión endógena (Isaacs et al., 2004).

En este último grupo se encuentra uno de los componentes más ampliamente utilizado para expresión genética en sistemas artificiales, correspondiente a un sistema proveniente del fago T7 de *E. coli* (Studier & Moffatt, 1986), el cual está compuesto por el par –mutuamente específico - polimerasa (RNAPT7) y promotor T7 (pT7). Gracias al desacoplamiento de la maquinaria de transcripción endógena, se genera una sobreexpresión de los genes bajo pT7 en presencia de la RNAPT7, con una mínima influencia del contexto. Este sistema ha sido optimizado y ampliado: se ha reducido la toxicidad de la RNAPT7, se han desarrollado variantes ortogonales (Temme, Hill, Segall-shapiro, Moser, & Voigt, 2012) y se ha separado la RNAPT7 en sub-unidades que necesitan ser co-expresadas para constituir la (Segall-Shapiro, Meyer, Ellington, Sontag, & Voigt, 2014), lo que aumenta la capacidad de control sobre el sistema y permite integrar diversas señales.

1.3 Simulación de sistemas multicelulares

Para poder abarcar la complejidad del diseño de sistemas multicelulares es fundamental el desarrollo y uso de modelos matemáticos y herramientas computacionales que permitan simular el comportamiento del sistema estudiado. Diversos esfuerzos se han realizado en torno a este desafío, por ejemplo, a partir de agentes celulares y sus interacciones espaciales, se han elaborado plataformas capaces de modelar el metabolismo, la señalización celular y la regulación

genética (Hoehme & Drasdo, 2010; Jang, Oishi, Egbert, & Klavins, 2012; Kreft, J. U., Booth, & Wimpenny, 1998). Con ellas se han definido reglas abstractas para la ingeniería de formas en sistemas biológicos, aunque estas aún no han sido implementadas *in vivo* (Pascalie et al., 2016).

Debido al número de células existentes en una colonia ($\sim 10^4$ - 10^7), la mayoría de los métodos de simulación presenta problemas al aproximarse a esa escala. En contraste, CellModeller (Rudge, Steiner, Phillips, & Haseloff, 2012) es capaz de simular el crecimiento de una colonia bacteriana de hasta 100.000 individuos a partir de una célula (Figura 1-2). CellModeller se basa en la simulación numérica multiparalela del crecimiento celular, utilizando OpenCL para abordar la computación paralela en la GPU de cada unidad celular como un cuerpo rígido. Las especificaciones sobre el comportamiento del sistema (escritas en *Python*) pueden ser dadas a través de reglas explícitas o ecuaciones diferenciales, con las cuales es posible describir la biofísica, la genética y la señalización de las células en crecimiento. En la Figura 1-3 se muestra el algoritmo central de su funcionamiento.

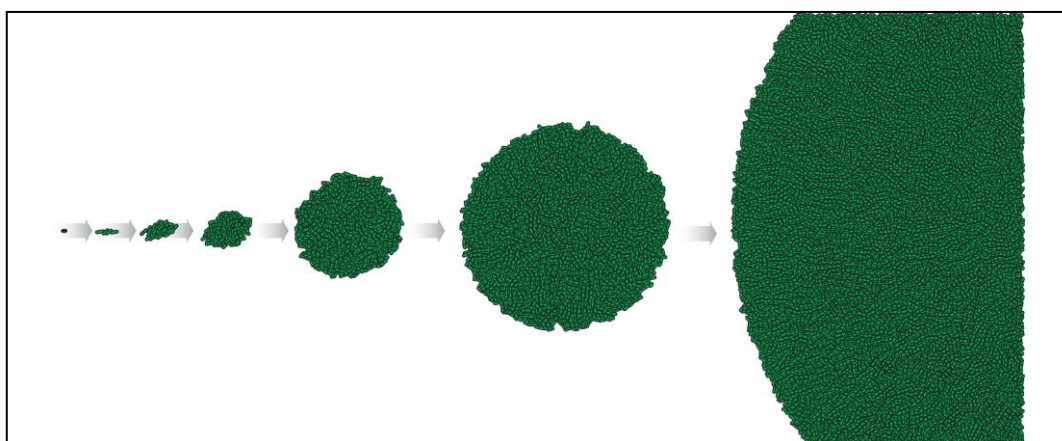


Figura 1-2: Simulación en CellModeller de colonia en crecimiento. Se observa como la simulación parte de una célula, y como esta forma una colonia mediante el proceso de replicación celular. Cada célula es un agente independiente (tamaño, posición, componentes y velocidad de crecimiento propios). * El estado final de la colonia fue recortado para el encuadre de la imagen.

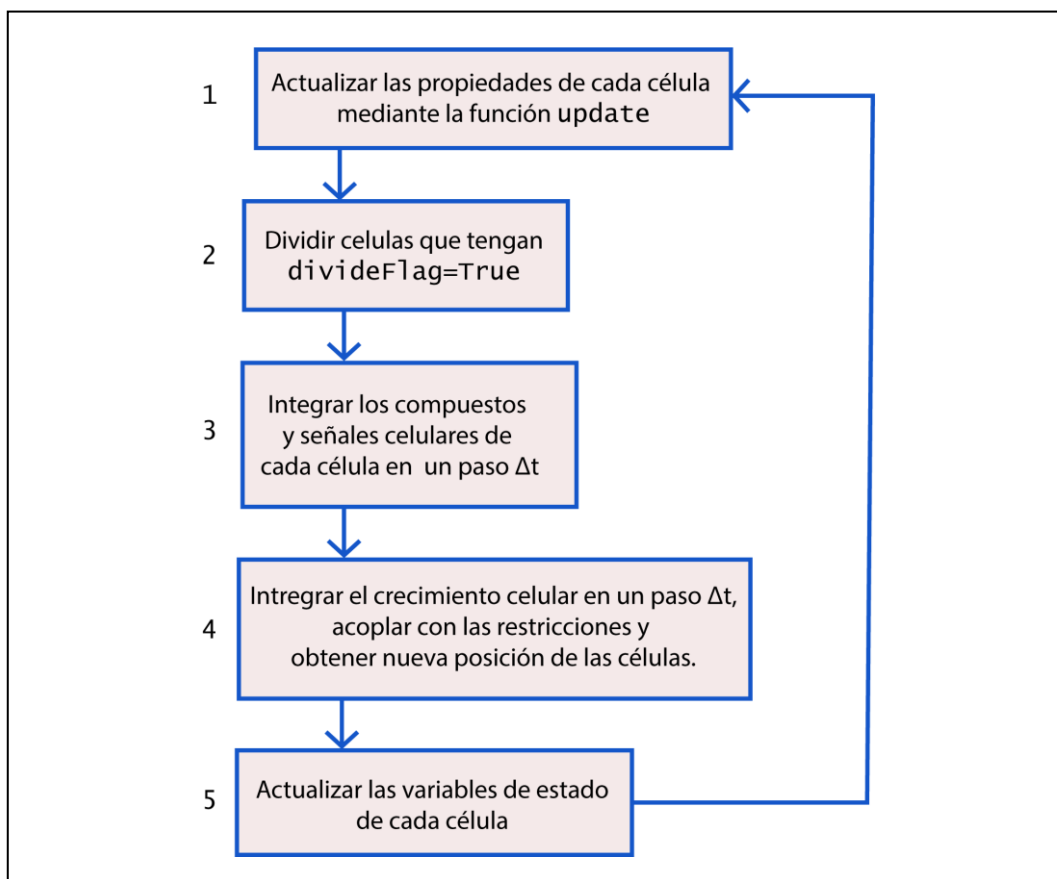


Figura 1-3: Esquema del algoritmo central de CellModeller. Cada cuadro representa una de las etapas secuenciales centrales llevadas a cabo en cada paso temporal de la simulación.

Debido a que el software está soportado en *Python* y es de código abierto (<http://haselofflab.github.io/CellModeller/>), este puede ser modificado por la comunidad de usuarios para satisfacer diversas situaciones y objetivos. Más aún, este aspecto permite su integración con las múltiples librerías existentes en este lenguaje, facilita la creación de nuevos módulos y la reutilización de otros ya desarrollados.

CellModeller ha sido útil en el estudio de interacciones físicas entre células de cultivos multicelulares (Rudge et al., 2013; Smith et al., 2016). En esta tesis se ha

utilizado CellModeller para guiar el diseño de funciones genéticas y explorar su impacto en la morfología de las colonias.

1.4 Sistema CRISPRi

El sistema CRISPR/Cas9 es un novedoso método de manipulación genética obtenido a partir del sistema inmune natural de las bacterias (Makarova et al., 2011). Este es capaz de asociar el complejo proteico Cas9 a una secuencia específica de ADN, mediante un RNA guía (sgRNA), y mutarlas en sitios adyacentes debido a su actividad *endonucleasa*, modificando de manera permanente la secuencia objetivo (Haurwitz, Jinek, Wiedenheft, Zhou, & Doudna, 2010).

Dicho sistema ha sido modificado para obtener CRISPRi, una versión de CRISPR para efectuar regulación de la expresión génica de manera transciente mediante la interferencia del proceso de transcripción (Larson et al., 2013; Qi et al., 2013). Para ello se modificó la proteína Cas9 a dCas9 eliminando su actividad *nucleasa*, y así poder regular la expresión genética simplemente por impedimento estérico del complejo dCas9-sgRNA sobre la enzima ARN polimerasa. Este modo de interacción permite bloquear la expresión de una secuencia de ADN sólo mientras se encuentra presente dicho complejo, siendo un proceso reversible.

Uno de los principales potenciales de esta técnica es la simplicidad, modularidad y versatilidad en el sistema reconocimiento de la secuencia objetivo. Este está determinado por la secuencia del sgRNA y restringido simplemente por una secuencia PAM (3 nucleótidos inmediatamente adyacentes al ADN target). De este modo, con el simple diseño de una cadena corta de ADN (que *in vivo* se transcribe a ARN), se puede programar el complejo para regular la expresión de secuencias genéticas arbitrarias presentes en plásmidos o en el genoma, sin necesidad de

modificarlos. Esto ha permitido, por ejemplo, la regulación de secuencias en el genoma para producir cambios morfológicos a nivel celular (Elhadi, Lv, Jiang, Wu, & Chen, 2016).

2 HIPOTESIS Y OBJETIVOS

2.1 Hipótesis

El uso de funciones biológicas elementales y abstracciones computacionales permiten obtener trazabilidad en la ingeniería de cambios morfogenéticos en colonias de *E. coli*.

2.2 Objetivo General

Diseñar un sistema genético modular que permita realizar ingeniería morfogenética en colonias de *E. coli* con un enfoque *bottom up*.

2.3 Objetivos específicos

Se plantearon los siguientes objetivos específicos:

- a) Diseñar una lógica genética modular para romper simetría y establecer subdominios celulares en colonias de *E.coli*.
- b) Utilizar el sistema diseñado para prototipar regulación celular dominio específico dentro de la colonia bacteriana.
- c) Representar estos módulos como procesos computacionales en CellModeller (Rudge et al., 2012) para guiar el diseño experimental y explorar el espacio de formas posibles.
- d) Integrar las funciones elementales generadas para instruir cambios metabólicos y/o morfológicos en colonias de *E.coli*.

3 METODOLOGIA

3.1 Ensamblaje y preparación de ADN

La construcción de todos los plásmidos fue llevada a cabo por ensamblaje basado en homología, utilizando la técnica *Gibson assembly* (Gibson et al., 2009). Por cada ensamblaje se realizó *colony* PCR a tres colonias, donde los partidores a utilizar fueron escogidos para cada caso de manera de chequear la inserción de todas las piezas en el *backbone*, en el lugar adecuado. Dependiendo del resultado anterior, hasta dos de los vectores positivos fueron enviados a secuenciar a MacroGen Inc. Korea © (<http://dna.macrogen.com/>). Los resultados de fluorograma y secuencia obtenidos fueron revisados por alineamiento con el programa de libre acceso *Ape* (<http://biologylabs.utah.edu/jorgensen/wayned/ape/>), en el cual también fueron realizados todos los diseños de secuencias y partidores.

Todos los partidores fueron sintetizados por IDT Inc., resuspendidos en agua de cromatografía a concentración stock de 100 μ M y diluidos a concentración de uso 10 μ M, siendo ambos almacenados a -20°C. Los PCR para la construcción de fragmentos de ADN fueron realizados con Phusion® High-Fidelity DNA Polymerase (NEB) de acuerdo con las condiciones indicadas por el fabricante, y luego resueltos en geles de agarosa 1% o 2% - para fragmentos mayores y menores a 500 bp respectivamente – con SYBR® Safe (ThermoFisher) para teñir el ADN. En ellos se seleccionaron y cortaron las bandas adecuadas, las cuales fueron purificadas con el kit Wizard® SV Gel and PCR Clean-Up System (Promega). Finalmente, las concentraciones obtenidas fueron cuantificadas mediante el sistema Take3® en un lector de placas Synergy HTX (BioTek).

Todos los vectores fueron clonados en *E. coli* Top10 (Invitrogen). Los plásmidos fueron extraídos mediante el kit Wizard® Plus SV Minipreps DNA Purification System (Promega). Se tomaron 500 μ l de cada cultivo previo a la purificación, los

cuales fueron mezclados con 500 µl de glicerol 50% v/v (Tabla 3.3) y guardados a -80°C en tubos criogénicos a modo de stock.

3.2 *Colony PCR*

Esta técnica fue utilizada para realizar *screening* de las colonias bacterianas candidatas luego de realizar ensamblaje de ADN y transformación. Este método evita el realizar cultivo y purificación de plásmido a todas las colonias testeadas, reduciendo el costo y tiempo de los experimentos.

Después de 12 -16 horas de realizada la transformación, las colonias candidatas fueron “picadas” con una pipeta y resuspendidas en 10 µl de H₂O (Merk ®), en condiciones de esterilidad. De ello, 2 µl fueron inoculados en un tubo bacteriano con LB fresco y el antibiótico apropiado. Por otra parte, 0,4 µl fueron utilizados para realizar PCR, para lo cual se mezclaron con 9,4 µl de mix de PCR y 1 µl de cada primer (10 µl volumen total). El programa del PCR se realizó de acuerdo con las especificaciones de la polimerasa utilizada, a excepción de que la elongación inicial se extendió por 10 min a 95°C para lisar las bacterias. Finalmente, los amplicones fueron chequeados por electroforesis en un gel de agarosa. Particularmente en este trabajo todos los *colony PCR* fueron realizados con *GoTaq® DNA Polymerase* (Promega).

3.3 *Condiciones de crecimiento*

Todos los experimentos fueron realizados a partir de la cepa *E. coli* Top10 (Invitrogen). La única excepción a esto, fueron los experimentos de segregación con 3 y 4 plásmidos, donde se utilizó la cepa *E. coli* DH5αZ1.

Los cultivos fueron realizados en LB (Tabla 3.1) o en M9 (Tabla 3.2), los cuales fueron suplementados con agar cuando se hicieron cultivos sólidos. Para los experimentos de auxotrofía en metionina, se preparó medio M9 sin casaminoácidos, pero suplementado con L-leucina (30 mg/ml) (Tabla 3.2). Los cultivos fueron suplementados con los antibióticos apropiados cuando fue necesario, utilizando una concentración final 1X de acuerdo con la Tabla 3.3. La temperatura de crecimiento siempre fue de 37°C, utilizando tubos *Falcon*™ *Round-Bottom* y agitación de 250 RPM para cultivos líquidos.

Tabla 3.1 : Receta medio LB

Componente	Concentración final
Triptona	10 g/L
Extracto de levadura	5 g/L
NaCl	5 g/L
Agar *	15 g/L

Las cantidades requeridas de cada componente fueron disueltas en agua destilada, y posteriormente autoclavadas. * Solo suplementado para medio sólido.

Tabla 3.2: Receta Medio M9

	M9	M9 – Leu
5X Sales M9	200 ml/L	200 ml/L
MgSO ₄ *7H ₂ O	2 mM	2 mM
CaCl ₂	0,1 mM	0,1 mM
Glucosa	0,4 g/L	0,4 g/L
Casaminoácidos	0,2 g/L	-

L-leucina	-	30 mg/L
Agar 3% *	500 ml /L	500 ml /L

Todos los compuestos fueron añadidos a partir de sus respectivas soluciones stock, en orden descendente de acuerdo a la tabla y siempre en esterilidad. * Solo suplementado para medio sólido.

Tabla 3.3 : Soluciones stock

	Componentes	Esterilidad	Almacenamiento
5X Sales M9	64 g/L Na ₂ HPO ₄ ·7H ₂ O 15 g/L KH ₂ PO ₄ 2.5 g/L NaCl 5 g/L NH ₄ Cl Disueltos en agua destilada	Autoclavado	Temp. ambiente
Agar 3%	30 g/L agarosa Disueltos en agua destilada	Autoclavado	Temp. ambiente
MgSO ₄ ·7H ₂ O 1M	1 mol MgSO ₄ ·7H ₂ O Disueltos en agua destilada	Autoclavado	Temp. ambiente
CaCl ₂ 1M	1 mol CaCl ₂ Disueltos en agua destilada	Autoclavado	Temp. ambiente
Glucosa 20% p/v	200 g /L Glucosa Disueltos en agua destilada	Filtrado en trompo 0,22 µm	Temp. ambiente
Metionina 1000X	12 g/L Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	4 ° C
L-leucina 500 X	15 g/L Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	4 ° C
IPTG 0,1 M	0,1 mol IPTG Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	-20 °C
L-Arabinosa 1M	1 mol L-Arabinosa Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	Temp. ambiente
Glicerol 50% v/v	500 ml/L glicerol Disuelto en agua destilada	Autoclavado	Temp. ambiente

Carbenicilina 1000X	100 g/L Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	-20 °C
Kanamicina 1000X	50 g/L Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	-20 °C
Cloranfenicol 1000X	10 g/L Disuelto en etanol absoluto (Merk)	Filtrado en trompo 0,22 µm	-20 °C
Tetraciclina 500X	5 g/L Disuelto en etanol absoluto (Merk)	Filtrado en trompo 0,22 µm	-20 °C
Espectinomicina 1000X	50 g/L Disuelto en agua de cromatografía (Merck)	Filtrado en trompo 0,22 µm	-20 °C

3.4 Transformación y Co-transformación bacteriana

Todas las transformaciones de 1 y 2 plásmidos (*i.e.* co-transformación) fueron llevadas a cabo en alícuotas de 50 µl de bacterias quimio-competentes de la cepa *E. coli Top 10* (Invitrogen). Estas fueron preparadas por el método CCMB80 de competencia (http://openwetware.org/wiki/TOP10_chemically_competent_cells), alicuotadas y almacenadas a -80°C. Al momento de efectuar la transformación, estas fueron descongeladas durante 20 min en hielo y luego mezcladas con 1 µl de ADN a ~1ng/µl. Para la co-transformación, se tomó un volumen de cada plásmido equivalente a 100 ng de él, se mezclaron y se completó con H₂O hasta tener un volumen total de 6 µl, los cuales fueron añadidos a las bacterias. Los tubos se mantuvieron en hielo por 25 min antes de llevar a cabo el *heat shock* por 45 segundos en un baño de agua caliente a 42°C. Las células se recuperaron durante 2 -5 min en hielo previo a añadir 250 µl de LB, y se dejaron incubando 1hr a 37°C con agitación a 250 RPM. A continuación, se plaquearon 25 µl en LB agar con el antibiótico apropiado. Para las co-transformaciones el tiempo de incubación ascendió hasta 1,5 h y se plaquearon 250 µl de las bacterias transformadas. Cuando la co-transformación contempló 2 o más plásmidos SEG# se suplementó con una

concentración de arabinosa 5 mM al momento de la incubación y en las placas. Los cultivos en LB agar fueron incubados *over night* (~12-16 h) antes de picar colonias. Cuando se requirió introducir 3 plásmidos en una bacteria, se realizaron transformaciones y co-transformaciones en serie.

Para la segregación de 3 y 4 plásmidos, las transformaciones fueron llevadas a cabo en bacterias quimio-competentes de la cepa *E.coli DH5 α Z1*, añadiendo los antibióticos en las concentraciones 1x a partir de lo indicado en la Tabla 3.3 y suplementadas con 1% arabinosa. La co-transformación de 4 plásmidos se llevó a cabo mediante el siguiente procedimiento secuencial:

- a) Transformación de cepa DH5 α Z1 quimio-competente con plásmido SEG10.
- b) Mediante procedimientos estándares, se prepararon células electro-competentes de las cepas obtenidas en (a).
- c) Las células preparadas en (b) fueron co-transformadas con SEG11 y SEG12, y recuperadas en medio SOC suplementado con arabinosa.
- d) Las células fueron cultivadas en placas con LB agar suplementadas con carbenicilina, tetraciclina, estreptomycin (250 μ g/ml) y arabinosa. Estas se incubaron a 37°C por 30 h para ver colonias.
- e) Se hicieron células quimio-competentes a partir de la cepa obtenida en (d).
- f) Las células preparadas en (e) fueron transformadas con SEG13, recuperadas en medio SOC con arabinosa y plaqueadas en medio LB sólido con los cuatro antibióticos (carbenicilina, tetraciclina, estreptomycin y kanamicina) más arabinosa.
- g) Se picaron colonias de (f), las cuales fueron cultivadas en medio LB líquido, con los mismos 4 antibióticos más arabinosa, a 37°C y agitación a 270 RPM.
- h) Cuando el cultivo de (g) alcanzó turbidez, se diluyó 1:1000 y se dejó creciendo hasta alcanzar turbidez nuevamente. Luego se realizó un ciclo más de dilución 1:1000 – alcanzar turbidez antes de guardar glicerol de la cepa con los 4 plásmidos SEG#.

3.5 Registro de imágenes

Todas las imágenes fueron tomadas por microscopia de fluorescencia o de campo claro en un equipo *Nikon Ni*, a excepción de la segregación de 4 plásmidos (detallado abajo). Los filtros utilizados para las proteínas fluorescentes de estas imágenes se presentan en la Tabla 3.4. El análisis de estas imágenes y la composición de los canales fue realizado en MATLAB® (códigos en Anexo B) para fines de intensidad y forma. El nivel de imagen y brillo fue ajustado para la imagen completa en Adobe Photoshop (CS6) cuando esta no tenía fines cuantitativos.

Tabla 3.4: Filtros de fluorescencia utilizados para cada proteína fluorescente

	Excitación [nm]	Emisión [nm]	Dicroico [nm]
mTurquoise	425 -445	465 -495	455
sfGFP	465 - 495	515 - 555	505
mVenus	490 - 510	520 - 550	515
mCherry	540 - 580	605 - 695	600

Las imágenes de segregación de 4 plásmidos fueron tomadas por Atri Choksi (Endry Lab, Standford University) con un microscopio *confocal inverted Zeiss LSM 780 multiphoton laser scanning*, usando un objetivo 10x. Las longitudes de onda utilizadas para la excitación y el rango de recepción se presentan en la Tabla 3.5. Para componer la imagen final con todos los canales de manera simultánea, se utilizó el *spectral unmixing plugin* de *ImageJ* y luego se hizo *merge* en el mismo (Schneider, Rasband, & Eliceiri, 2012).

Tabla 3.5: Especificaciones de las longitudes de onda utilizadas en la segregación de cuatro plásmidos.

	Excitación [nm]	Emisión [nm]
mTurquoise	458 (argón ion laser)	463 - 581
sfGFP	488 (argón ion laser)	499 - 580
mVenus	488 (argón ion laser)	526 - 598
mCherry	561 (HeNe laser)	599 - 697

3.6 Análisis de imágenes en ensayo de cambios morfológicos

Todas las imágenes tratadas para fines cuantitativos y los análisis descritos en esta sección fueron realizados en MATLAB®. Los códigos utilizados para este propósito fueron desarrollados en esta tesis y se adjuntan en el anexo B.

Las imágenes registradas por microscopía (ver sección 3.5) fueron convertidas a formato RGB mediante la rutina *convertSingleImagesToRGB.m*. A continuación, la composición de estas fue realizada con la rutina *mergeImagesToRGB.m*, mediante la simple adición de los valores de ambas imágenes para cada canal y guardado como un nuevo archivo RGB.

Mediante la rutina *ColonySelection.m*, se seleccionó de forma manual la colonia de interés en cada una de las imágenes compuestas, haciendo *click* en un punto al interior de esta (registrado en el archivo *ColPos.mat*). Luego, se ejecutó el código *AllParameters.m*, con el cual: se suavizaron las imágenes con un filtro gaussiano (*fspecial*, *Image Processing Toolbox*), se segmentó cada colonia del resto de la imagen en base a un valor de intensidad umbral para definir sus límites, y se

computaron valores para caracterizar su morfología (excentricidad, circularidad, etc).

Tanto los parámetros que definen el filtro gaussiano aplicado (tamaño del filtro – en píxeles- y desviación estándar (σ)), así como los valores umbral de cada dominio de la colonia, fueron seleccionados en base a criterios visuales con la ayuda de la función *getParameters*. Estos criterios involucraron la disminución del ruido, mantener la suavidad y continuidad del borde de la colonia, y el preservar la estructura de ella respecto a la imagen original. Los valores escogidos para cada imagen se almacenaron en el archivo *Parameters.mat*.

Todos los cálculos se efectuaron en base a la región segmentada como parte de la colonia de interés. Mediante el *Image Processing Toolbox*, se obtuvo directamente el área, perímetro y excentricidad de cada colonia (almacenados en *MorphVal#h.mat*). La ANOVA y el test multi-comparativo de los valores obtenidos se realizaron mediante las funciones *anova1* y *multcompare* del *Statistics and Machine Learning Toolbox*. Este análisis estadístico y las figuras realizadas se encuentran en la rutina *MorphStatic.m*.

3.7 Ensayos de cepas auxótrofas en metionina e inducción de cambios morfológicos

Se cultivaron colonias individuales transformadas con S11Ma + SEG8 + pAct6 en medio LB suplementado con kanamicina, tetraciclina, carbenicilina y arabinosa (5 mM), a 37°C con agitación a 250 RPM (etapa de propagación). Después de 5 horas de cultivo, las células fueron centrifugadas a 3000 RPM durante 10 minutos para formar un pellet. Este fue resuspendido en 5 ml de medio LB fresco con cloranfenicol y kanamicina (etapa de decaimiento del número de copias). Estas

fueron diluidas inmediatamente a razón 1:5000 con el mismo medio y se plaquearon 10 μ l de dicha dilución en cada placa de medio mínimo M9 sin casaminácidos. En ellas se realizaron 4 tratamientos: placas suplementadas con metionina (12 mg/l) más IPTG 0,2 mM, placas suplementadas solo con IPTG (inducción a $t=0$ h), placas suplementadas con IPTG después de 8 horas de plaquear (inducción $t=8$ h) y placas sin IPTG ni metionina (grupo control). A las 24 horas de cultivo se aplicó un refuerzo de IPTG en las placas que involucraban tratamiento con él. Todas las aplicaciones post-cultivo (*i.e.* inducción a las 8 h y refuerzo) se realizaron mediante aplicación directa de 30 μ l IPTG 0,1 M (equivalente a una concentración final de 0,2 mM) distribuidos en varios puntos de manera homogénea en la placa.

3.8 Implementación de módulos en CellModeller

El módulo *Python* (Anexo C) desarrollado para representar la segregación de plásmidos en CellModeller comienza con una célula inicial que contiene un número N de cada uno de los plásmidos (*i.e.* para dos plásmidos el total dentro de ella será $2N$). Luego, en el paso de división células (paso 2, Figura 1-3) los plásmidos se duplican y son heredados de forma aleatoria, y en igual número, a cada una de las células hijas.

Por su parte, el módulo que representa el crecimiento celular diferenciado en cada dominio de la colonia se implementó directamente vía la tasa de crecimiento, que se actualiza en la función *update* (paso 1, Figura 1-3). Este parámetro se hizo variar entre 0,5 y 1,0 de acuerdo con el número de plásmidos inhibidores del crecimiento dentro de la colonia. Finalmente, para representar la inducción temporal de la regulación del crecimiento, el módulo anterior se situó dentro de

una sentencia condicional, que solo se ejecuta al superar un número crítico de células definido por el usuario (Figura 3-1).

```
def update(cells):
    for (id, cell) in cells.iteritems():
        if len(cells)>1000:
            gr1 = 1.0
            gr2 = 0.5
            cell.growthRate = (gr2-gr1)*cell.n_a*0.5 + gr1

            cell.color = [0.1, cell.n_a/2.0, cell.n_b/2.0]
        if cell.volume > cell.targetVol:
            cell.divideFlag = True
```

Figura 3-1: Módulo *update* implementado en simulaciones. En la figura se muestra el módulo *update* con valores representativos. Para este caso, la sentencia condicional solo se ejecutará cuando la colonia comprenda más de 1000 células, momento en que se modifica la tasa de crecimiento y el color con que está marcada cada célula de acuerdo con el número de cada tipo de plásmido que tenga en su interior. Finalmente, se observa el módulo para actualizar el estado de división celular en base a si su volumen ha superado el tamaño critico definido.

3.9 Caracterización de la segregación de plásmidos SEG#

Para los análisis inducción del número de copias de plásmidos SEG# ante arabinosa, se utilizaron cultivos *overnight* de SEG3 y SEG4. Estos fueron diluidos en LB con cloranfenicol más el suplemento apropiado de arabinosa, y crecidos con agitación en un lector de placas (Clariostar plate reader, BMG LABTECH) durante 16 horas con mediciones cada 10 minutos de absorbancia (600 nm) y fluorescencia (mCherry y sfGFP). Los datos presentados corresponden al promedio de la tasa de fluorescencia por OD $[(dF/dt)/OD]$ a lo largo de la fase exponencial de crecimiento.

Para la caracterización del efecto de la suplementación con arabinosa en el rendimiento de ADN de bacterias transformadas con plásmidos SEG#, se

utilizaron cultivos *overnight* de SEG3 y SEG4. Estos correspondieron a 5 ml de LB suplementados con cloranfenicol y la cantidad apropiada de arabinosa. En el lector de placas se midió la densidad óptica (600 nm) de dichos cultivos y se llevó a cabo el procedimiento de *miniprep* de acuerdo con las instrucciones del fabricante (QiaPrep Spin Miniprep Kit, Qiagen), cuya concentración de ADN fue medida en un *NanoVue Plus* (GE). Los resultados presentan el rendimiento de DNA total por OD por ml de cultivo [ng DNA/(OD*ml)].

Para la caracterización de como la densidad de cultivo genera choque de colonias con apariencia de bipartitas, se realizaron cultivos *overnight* en 5 ml de LB líquido, suplementado con cloranfenicol y arabinosa (5 mM), de las cepas SEG3 y SEG4. La densidad óptica (600 nm) fue medida para ambos cultivos y en base a ello se realizó una mezcla de 1 ml con mismo número de células de cada cepa. Un medio LB fresco suplementado con cloranfenicol y arabinosa se utilizó para realizar una dilución 1:1000 de la mezcla. Esta fue cultivada a 37°C con agitación durante 5 horas ($OD_{600} \cong 0.08$), momento en el que se realizaron diluciones seriadas entre 1:1 y 1:100.000, cultivando 35 μ L de cada una en placas de LB agar suplementado con cloranfenicol. Las imágenes fueron tomadas por microscopía después de 24 horas de incubación a 37°C, y se realizó el conteo total de colonias directamente en las placas.

Los ensayos de segregación del resto de combinaciones de plásmidos SEG#, fue llevada a cabo a partir de cultivos líquidos *overnight* hechos directamente de placas frescas de colonias co-transformadas. Los cultivos se realizaron en tubos bacterianos con 5 ml de LB, los antibióticos apropiados para cada plásmido y arabinosa (5 mM). Estos fueron diluidos 1:1000 con un medio fresco de las mismas características e incubados por 5 horas a 37°C con agitación a 250 RPM. Luego de esto, los tubos fueron centrifugados 10 minutos a 3000 RPM para formar un pellet, el cual fue resuspendido en medio fresco suplementado con

cloranfenicol e incubado por 3 horas a 37° con agitación a 250 RPM. Transcurrido este tiempo, se hicieron diluciones 1:5000 con LB líquido y se cultivó 30 µL en cada placa de LB sólido con cloranfenicol.

El ensayo para la segregación de cuatro plásmidos fue realizado por Atri Choksi (Endry Lab, Stanford University) y se hizo a partir de un stock de glicerol de células DH5αZ1 transformadas con los 4 plásmidos (sección **¡Error! No se encuentra el origen de la referencia.**), el cual fue inoculado en medio LB con los cuatro antibióticos respectivos más arabinosa 1%. Se cultivó *overnight* a 37° con agitación a 270 RPM, para luego ser diluida 1:1000 con medio fresco de la misma composición, pero sin arabinosa y cultivado en idénticas condiciones durante 3 horas. Luego, las células fueron inoculadas en *pads* de agarosa siguiendo el protocolo de Young et al. (2012) a una densidad óptica OD600 = 0.001. Los *pads* fueron incubados a 37°C por 24 horas y luego a 25°C por otras 24 horas antes de obtener las imágenes.

4 RESULTADOS

4.1 Abstracción y desacoplamiento

Para poder obtener una plataforma simple de prototipado de mecanismos morfogénéticos - capaces de comenzar desde una célula individual y controlar la morfología a nivel multicelular -, se utilizaron colonias bacterianas crecidas en medio sólido desde células individuales (*colony forming units*, CFU) (Figura 4-1, A). En base a su proceso de crecimiento, la generación de formas en colonias bacterianas fue desacoplada en dos funciones elementales abstractas: i) Ruptura de simetría ii) Regulación celular dominio específica (Figura 4-1, B). El primer paso permite establecer polaridad y dominios separados en la colonia, y el segundo, regular procesos celulares en dichos sectores de manera diferenciada. Este enfoque modular fue diseñado para obtener una mayor trazabilidad en el ciclo de diseño-fabricación-testeo de funciones morfogénéticas, generando resultados más parcelables e informativos.

La lógica implementación de dichas funciones abstractas fue inspirado en la organización del sistema ABC de desarrollo de órganos florales, el cual explota la co-existencia espacial de reguladores transcripcionales para instruir la formación de sépalos (homodímeros AA), pétalos (heterodímeros AB), estambres (heterodímeros BC) y carpelos (homodímeros CC) (Bowman, Smyth, & Meyerowitz, 2012). Así, la superposición de dominios de expresión, *e.g* B + C, dan lugar a distintos órganos (Figura 4-1, C). En nuestra propuesta esto se implementó como sistema genético de dos niveles: uno para establecer dominios y polaridad, y otro para controlar procesos celulares, de cuya combinación es posible instruir comportamientos complejos específicos (Figura 4-1, D). Esta modularidad en niveles permite depurar el funcionamiento de los elementos de cada capa de manera independiente, y otorga flexibilidad combinatorial de los plásmidos y funciones desarrollados para cada una de ellas.

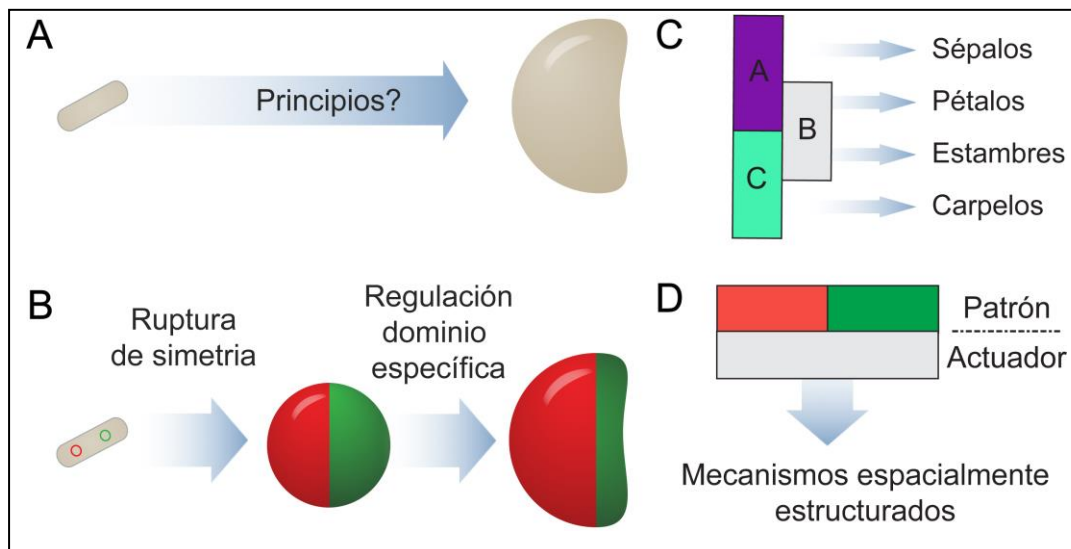


Figura 4-1: Desacoplamiento de mecanismos morfogénéticos en funciones elementales. A) Esquema del desafío de diseño morfogénético. B) Desacoplamiento en funciones elementales para abordar el desafío de manera trazable. C) Organización del modelo ABC de coexistencia espacial de factores de transcripción ('A', 'B' y 'C') en la formación de órganos florales (compuestos por sépalos, pétalos, estambres y carpelos). D) Esquema de la lógica en dos capas propuesta para implementar el sistema genético de las funciones desacopladas.

4.2 Ruptura de simetría y capa formadora de patrones

La ruptura de simetría es de suma importancia para la ingeniería de procesos concebidos a partir de una célula fundadora o sistemas homogéneos. La reducción de dicha homogeneidad genera polaridad, aumento en la estructura, y distribuciones espaciales que pueden servir como punto de referencia para otros procesos, *e.g.* funciones dependientes de gradiente. Este concepto representa la piedra angular para la formación de estructuras superiores en sistemas homogéneos, debido a que cada zona diferenciada puede adquirir un grado de especialización: componentes, respuestas y funciones determinadas (Anderson, 1972).

4.2.1 Diseño de la capa formadora de patrones

Para implementar esta primera función elemental se modificó un sistema genético artificial que permite marcar linajes celulares dentro de colonias bacterianas en crecimiento (Rudge et al., 2013). En él, los plásmidos que lo componen permiten identificar diferentes dominios en la colonia a partir del crecimiento de una célula fundadora. Esto ocurre debido a que dicha célula inicialmente contiene dos variantes de un plásmido, los cuales, siguiendo un proceso aleatorio, segregan en la medida que esta se multiplica.

El proceso se fundamenta en el diseño de los plásmidos (Figura 4-2), que contienen: i) genes de resistencia a antibióticos diferenciados (*i.e.* “A1^R” y “A2^R”), ii) expresión de proteínas fluorescentes con espectro separado y iii) tanto un gen de resistencia a antibiótico (“A3^R”) como replicón en común. Cuando una cepa co-transformada con ambos plásmidos es cultivada en presencia de los antibióticos A1 y A2, los dos vectores genéticos deben ser mantenidos en la célula para asegurar su supervivencia y el número total de copias asociado al replicón común es repartido entre ellos. Si posteriormente la célula es transferida a un medio con el antibiótico A3 solamente, ya no es necesario que la célula posea las dos resistencias a antibióticos para sobrevivir, haciendo que los plásmidos sean indistinguibles en este sentido, y generando que solo uno de ellos perdure en cada linaje de la colonia en crecimiento, de manera aleatoria e irreversible. Esta formación de linajes diferenciados se fundamenta en las propiedades mecánicas del crecimiento en colonias bacterianas (radial), lo que reduce la mezcla de sectores en torno al eje theta (Figura 4-3).

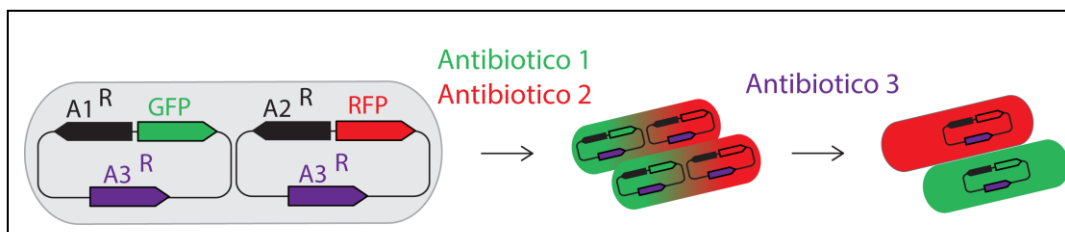


Figura 4-2: Mecanismo de segregación celular. La célula fundadora posee dos plásmidos distintos con resistencia a antibióticos A1 y A2, así como expresión de fluorescencia diferenciada. Al cultivar en medio con ambos antibióticos los dos plásmidos deben ser mantenidos en la célula, pero al cambiar a un medio de crecimiento que contiene únicamente el antibiótico A3 la célula solo necesita uno de los dos para sobrevivir y se produce la segregación celular.

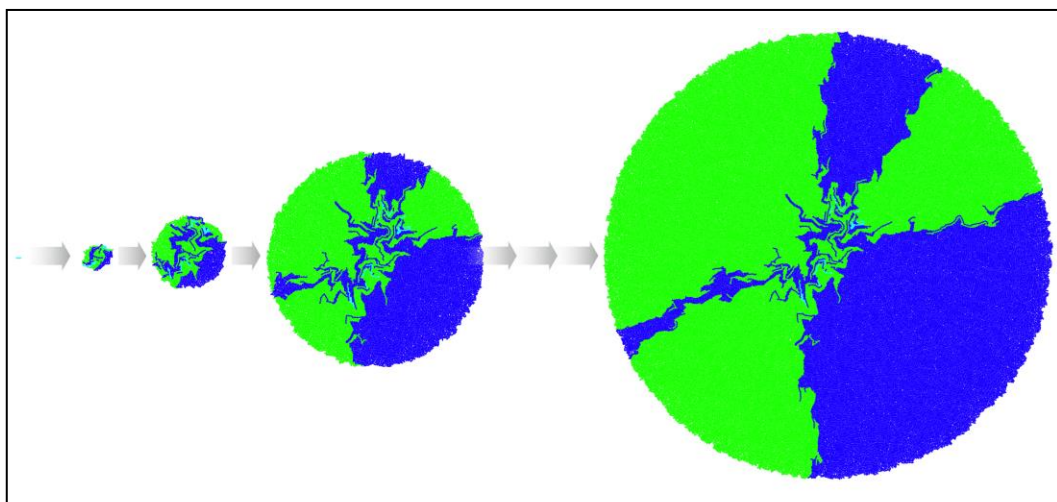


Figura 4-3: Simulación de colonia bacteriana en crecimiento. En la simulación se muestra el incremento radial de la colonia y la disminución de su mezcla en torno al eje theta en los estados más tardíos.

Sin embargo, el sistema original genera múltiples linajes celulares y no es útil para crear un número reducido de dominios más grandes. Desde el punto de vista de la ingeniería de la morfogénesis es interesante formar un número bajo y definido de sectores; en particular, lograr colonias bipartitas (*i.e.* dos dominios). Esto representa el punto de partida de este proyecto, debido a su simplicidad y homología con el proceso de polaridad en la embriogénesis. En esta tesis se re-

diseñó el sistema descrito en Rudge et al. (2013), para optimizar la obtención de solo un sector por plásmido en la colonia bacteriana.

4.2.2 Implementación en modelo biofísico

Para guiar el proceso de diseño se implementó un módulo de segregación en *CellModeller* (código en anexo C), con el cual se exploró el efecto del número de copias de plásmidos en la formación de sectores. Con ello se llevaron a cabo simulaciones con células portando 10 y 4 copias de plásmido en su interior, repartidas equitativamente, lo que solo generó colonias multi-sector (Figura 4-4 A y B). Para testear esto, se llevaron a cabo pruebas de segregación utilizando un plásmido con el origen de replicación pSC101* de baja copia (~4 a 10 copias/célula) (Lou et al., 2012), y tal como fue sugerido por las simulaciones, se obtuvieron únicamente colonias multi-sector (Figura 4-4 C).

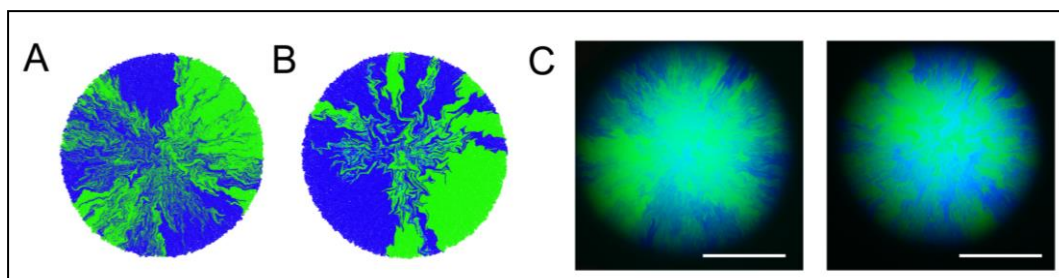


Figura 4-4: Colonias bacterianas multisector para 4 a 10 copias por célula. A) y B) corresponden a imágenes representativas de las simulaciones realizadas en *CellModeller* para 10 y 4 copias de plásmido por célula respectivamente. C) Imágenes representativas de colonias segregantes con dos variantes del plásmido psc101* capaces de expresar *sfGFP* (verde) y *Turquoise2* (azul) después de 24 h de crecimiento. Barra = 500 μm .

Para mejorar la comprensión del sistema, se entendió la herencia de plásmidos como un proceso binomial, supuesto bajo el cual fue posible calcular y simular la

probabilidad de obtener colonias bipartitas (Tabla 4.1). Estos resultados sugieren utilizar un ideal de 2 copias de plásmido en la célula fundadora, lo cual permitiría optimizar el proceso de formación de colonias con solo dos sectores para obtenerlas con un ~33% de probabilidad.

Tabla 4.1: Probabilidad de obtener colonias bipartitas a partir de diferente número de copias de plásmidos en célula fundadora.

Número de copias	Resultado	Esperado
2	0,3100	0,3333
4	0,0600	0,0285
6	0,0000	0,0021

En ella se contrasta el resultado obtenido al correr 100 simulaciones para cada caso, con el valor esperado de acuerdo con un modelo binomial de proceso.

4.2.3 Establecimiento de dominios celulares mediante ruptura de simetría

De acuerdo a lo sugerido por las simulaciones, se construyó un vector a partir del plásmido pDestBact (Torella, Boehm, et al., 2014) como base para los sistemas genéticos formadores de dominios (Figura 4-5 A). Este plásmido tiene la particularidad de poseer número de copias regulable desde 1 hasta ~40 mediante inducción con L-arabinosa. Para caracterizar esta propiedad se desarrollaron dos versiones de este plásmido: una con expresión de GFP y otra RFP, las cuales fueron sometidas a pruebas de inducción. De los resultados para producción de plásmido y generación de fluorescencia (Figura 4-5 B y C), se puede observar que la curva sigmoidea de inducción alcanza su valor máximo a partir de una

concentración de ~ 1 - 10 mM de arabinosa (Datos obtenidos por Anton Kan - University of Cambridge).

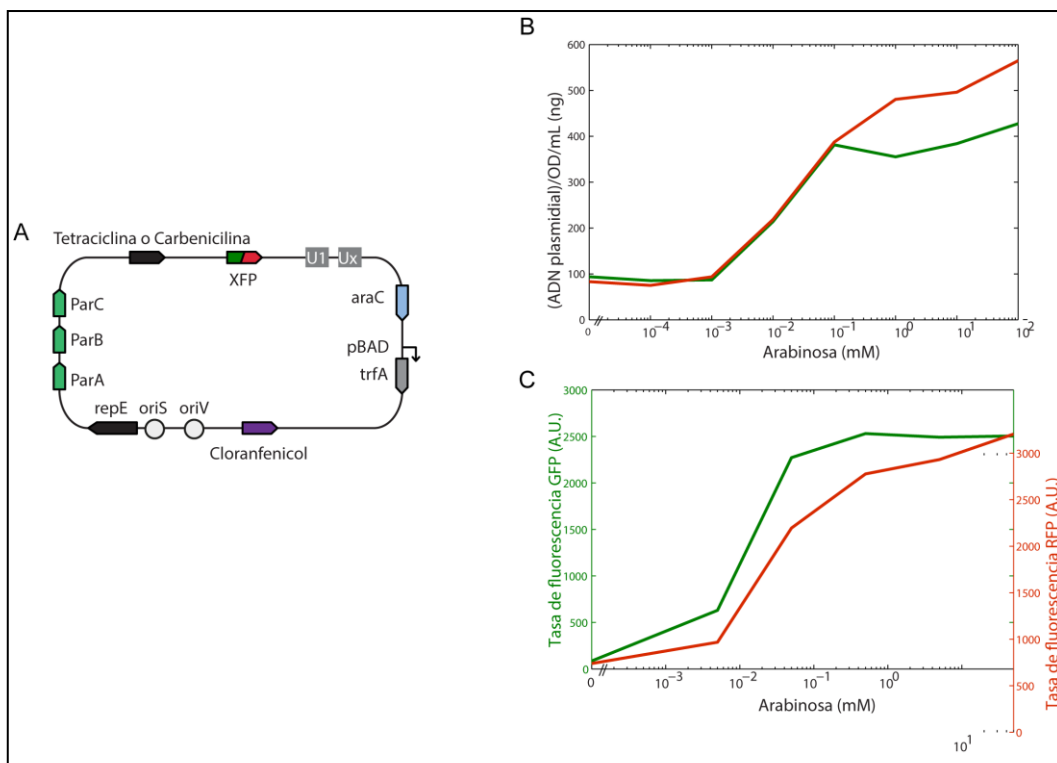


Figura 4-5: Plásmido desarrollado con número de copias inducible por arabinosa. A) Esquema del vector desarrollado a partir de pDestBact, este incluye: un sistema de replicación de baja copia (*ParABC*, *trfA*, *repE*, *oriS* y *oriV*) que es regulado por arabinosa (*pBAD/araC*), genes de resistencia para cloranfenicol y tetraciclina o carbenicilina, secuencias únicas para ensamblaje combinatorio (*U1,Ux*) y un gen de proteína fluorescente (*XFP*). B) y C) Resultados de inducción del plásmido desarrollado con dos proteínas fluorescentes diferentes (GFP en verde y RFP en rojo). Arriba se observa el rendimiento de producción de plásmido y abajo las mediciones de la tasa de fluorescencia producida por cultivos bacterianos en medio líquido. Datos de B) y C) obtenidos por Anton Kan (University of Cambridge).

En condiciones de copia media (*i.e.* inducción con arabinosa) es posible co-transformar bacterias con variantes del plásmido portando distintas resistencias a antibiótico ($A1^R$ y $A2^R$) y mantenerlas en la célula en cultivos suplementados con ambos antibióticos (“Fase propagativa”). Luego, se puede pasar una fase de “decaimiento” del número de copias plásmidiales mediante el cambio de medio cultivo a uno que no contenga el inductor y solo posea el antibiótico para el cual ambos plásmidos expresan resistencia ($A3$). Después de un determinado periodo de decaimiento la mayoría de las bacterias habrá reducido su número de plásmidos a solo dos, siendo el momento propicio para transferir el cultivo líquido a uno sólido (mismas condiciones del decaimiento) para comenzar la fase de “segregación” (Figura 4-6 A).

Para encontrar el punto óptimo en la obtención de bacterias fundadoras de colonias bipartitas, se realizaron dos experimentos de barrido de tiempo de cultivo en fase de decaimiento. Con ello, como se puede observar en la (Figura 4-6 B, el tiempo que optimiza la proporción de obtener dos sectores por colonia se encuentra en torno a los 180 minutos de cultivo en decaimiento previos a la fase de segregación (Datos obtenidos por Ilenne del Valle - PUC).

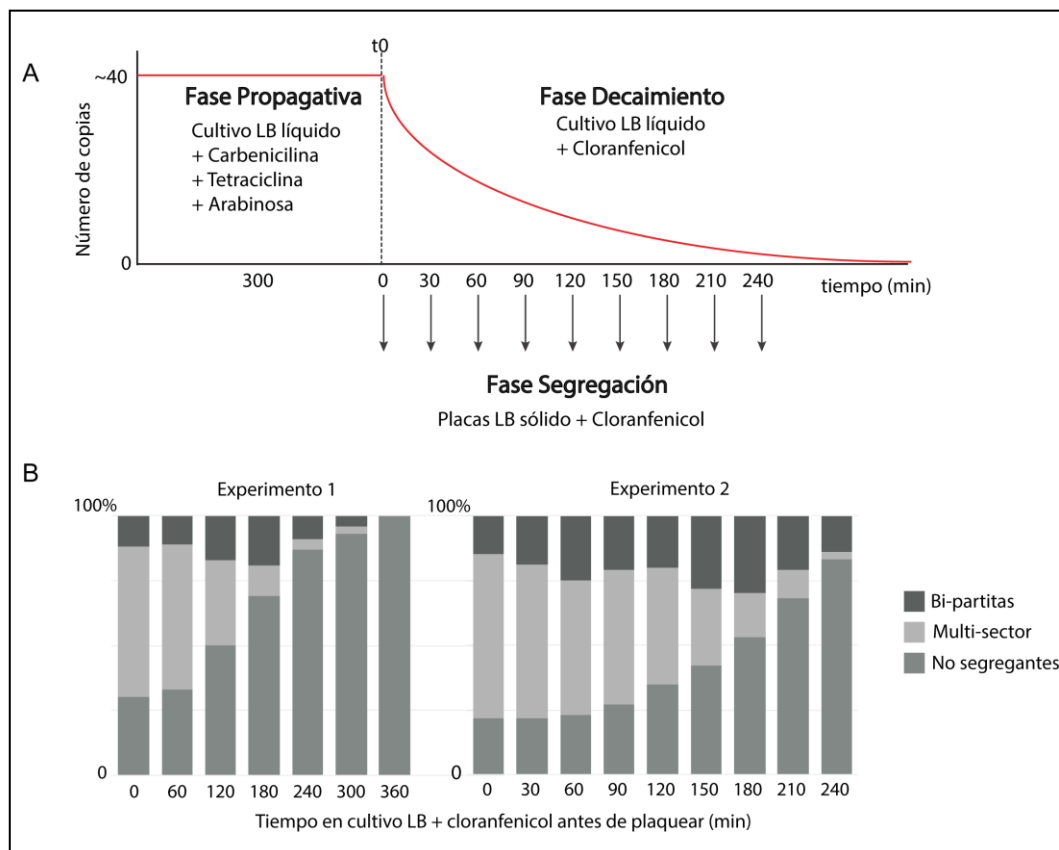


Figura 4-6: Fases y optimización de protocolo segregación. A) Especificaciones de las distintas fases del protocolo de segregación, en cada una de la cuales se produce un cambio de medio. B) Dos experimentos independientes para encontrar el óptimo para pasar a fase de segregación. Cada barra es un tiempo de cultivo en fase de decaimiento y en tonos de gris se indica la proporción de colonias de cada tipo a las 24 h de incubación. Datos obtenidos por Ilenne del Valle (PUC)

Las colonias bipartitas no solo se producen como resultado del proceso de segregación a partir de una célula fundadora, sino que también pueden ser observadas debido al choque de dos colonias bacterianas que comenzaron en puntos muy cercanos. Para cuantificar y evitar este tipo de “falso positivo”, se plaquearon distintas densidades celulares de mezcla de cultivos puros y se analizó su capacidad de formar colonias con dos sectores (Figura 4-7). De acuerdo con ello, cuando el cultivo en fase segregante tiene una densidad mayor a 7 colonias/cm² se producen falsos positivos. En vista de este resultado, al momento

de realizar el protocolo de segregación, se deben hacer diluciones de manera de obtener una densidad celular menor a dicho valor. Específicamente, en esta tesis se optó por llevar a cabo diluciones que conllevaran densidades celulares en torno a $\sim 1\text{-}3$ colonias/ cm^2 . Datos obtenidos por Ilenne del Valle (PUC).

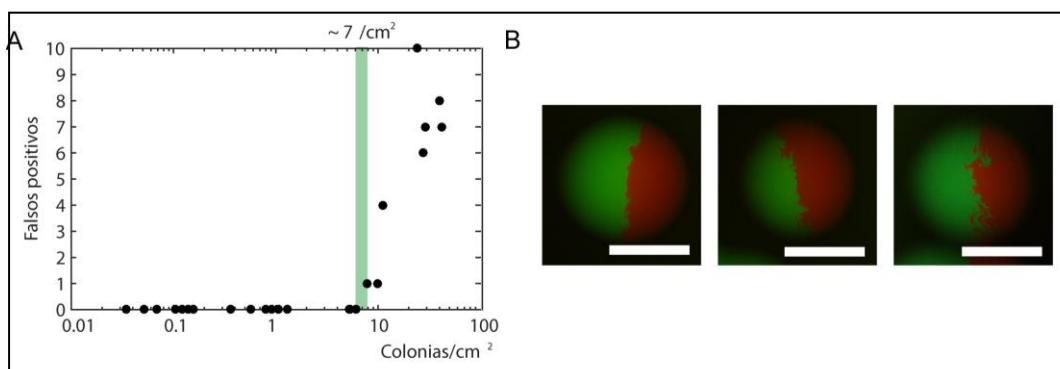


Figura 4-7: Falsos positivos. A) Resultado de mezclar dos cultivos con expresión de una proteína fluorescente y plaquearlos a distintas densidades. Al superar dicho valor las ~ 7 colonias/ cm^2 se puede observar la formación de colonias bipartitas falsas. B) Imágenes ejemplares de dicho comportamiento. Barra = 500 μm . Datos obtenidos por Ilenne del Valle (PUC)

Finalmente, para aumentar el espectro de diseño, se ensambló una serie de vectores genéticos (SEG#) con capacidad de expresar distintas combinaciones de resistencias a antibióticos y proteínas fluorescentes (sfGFP, mTurquoise2, mVenus, mCherry y “Blancos” sin marcador fluorescente). Co-transformando células con estos vectores (materiales y métodos), y siguiendo la lógica del protocolo de segregación definido, se establecieron diversas combinaciones de dos, tres y hasta cuatro sectores diferenciados dentro de una sola colonia (Figura 4-8). Adicionalmente, en la conformación de los plásmidos ensamblados se incluyeron secuencias nucleotídicas únicas “UNSeS” (Torella, Lienert, et al., 2014), que facilitan el ensamblaje combinatorial de unidades transcripcionales

flanqueados por estas secuencias en la posición U1 –UX dentro de los vectores diseñados.

Estos resultados muestran que se logró crear un sistema genético y un procedimiento capaz de formar hasta 4 dominios diferenciados dentro de una colonia bacteriana, de manera autónoma, a partir de una célula fundadora.

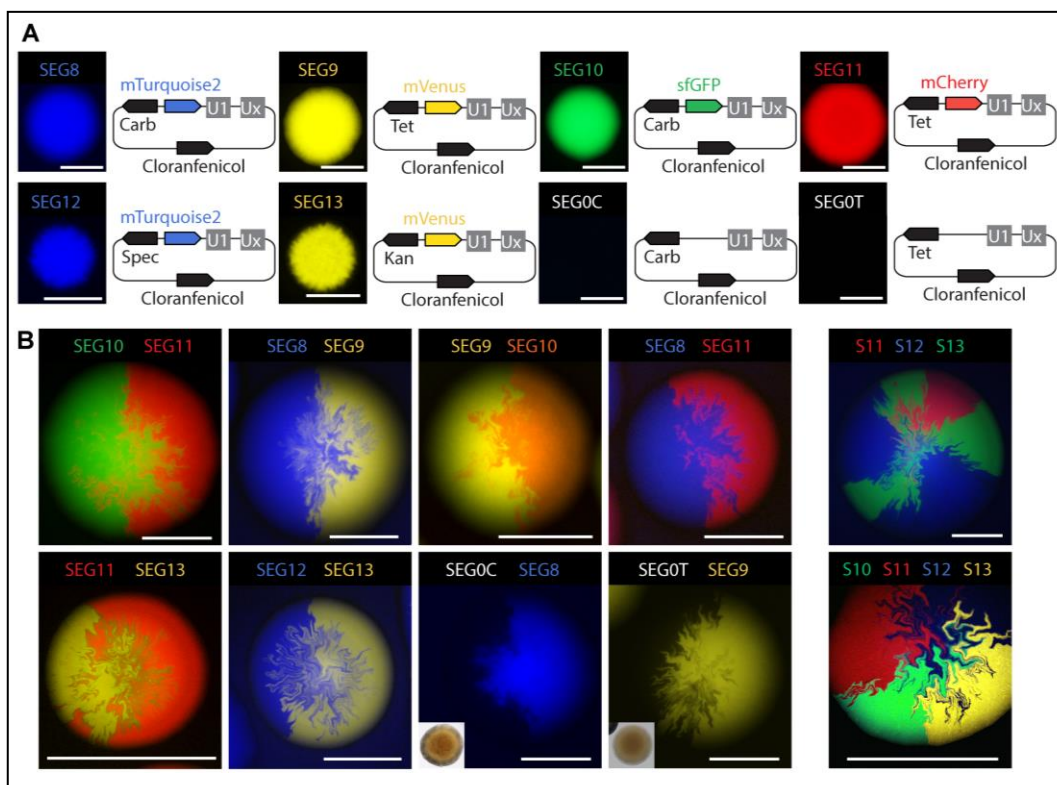


Figura 4-8: Serie de vectores “SEG” diseñados y definición de hasta cuatro sectores dentro de una colonia bacteriana. A) esquema de los diversos plásmidos “SEG” y sus especificaciones (U1 – UX corresponden a las secuencias “UNS”). Además, se presenta la imagen de una colonia representativa transformada con dicho vector. B) Imágenes de microscopía de fluorescencia de colonias segregantes con diversas combinaciones y cantidades de plásmidos SEG. Cada sector está asociado a un plásmido diferente. Barra = 500 μ m. Carb = gen resistencia a carbenicilina; Tet = gen resistencia a Tetraciclina; Kan = gen resistencia a Kanamicina; Spec = gen de resistencia a espectinomomicina. Data de colonia con 4 sectores obtenida por Atri Choksi y Drew Endy (Stanford University).

4.3 Capa actuadora y control de expresión genética dominio específica

De acuerdo con la lógica modular planteada para obtener regulación funcional dominio específica (Figura 4-1), fue necesario establecer una capa con localización ubicua en la colonia (*i.e.* está presente en cada célula de la colonia). Esta capa actuaría en respuesta o en conjunto a los elementos dominio-específico generados por la capa formadora de patrones. En base la interacción dichos elementos, se actuará sobre las funciones celulares con un control espacial y temporal.

Para validar la utilidad del sistema, versatilidad y efectividad del desacoplamiento modular, se utilizaron los plásmidos creadores de dominios y una serie de plásmidos actuadores ubicuos. A continuación se describe el control sobre la especialización de los dominios espaciales generados en la población mediante complementación metabólica y activación genética con sistema de transcripción basado en la ARN polimerasa T7 (RNAPT7) (Segall-Shapiro et al., 2014).

4.3.1 Ingeniería de la capa actuadora

Para este propósito se diseñó el plásmido ubicuo pAct, el cual contiene un gen de resistencia a kanamicina para asegurar su permanencia en el microorganismo, así como un origen de replicación de baja copia (psc101*) (Lou et al., 2012) compatible con el presente en los plásmidos SEG#. Para potenciar su modularidad, este vector también incluyó las regiones UNS, que permiten utilizarlo como base para el ensamblaje combinatorial de librerías de unidades transcripcionales.

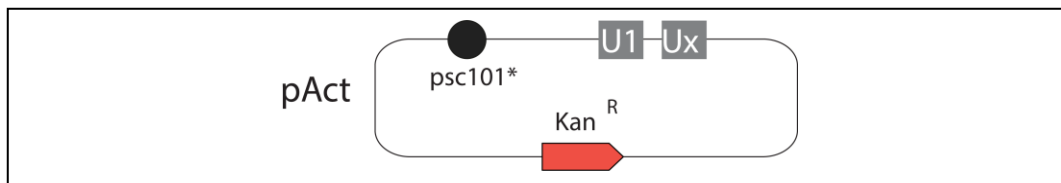


Figura 4-9: Esquema del plásmido pAct. Kan^R: gen de resistencia a kanamicina. U1, UX: Secuencias UNS para ensamblaje combinatorial de unidades transcripcionales. psc101*: Origen de replicación.

Como primer paso para probar el desempeño y ubicuidad en la expresión de este plásmido actuador, se le insertó una unidad transcripcional capaz de expresar la proteína fluorescente mVenus. En la Figura 4-10 se aprecia el resultado de un ensayo de segregación sobre esta nueva capa ubicua. Los plásmidos SEG8 y SEG11 expresaron de forma localizada la fluorescencia azul (mTurquoise2) y roja (mCherry) respectivamente, mientras que pAct1 expresó la fluorescencia amarilla (mVenus) de manera ubicua en toda la colonia. Estos resultados demuestran la capacidad de superposición modular de reglas de expresión genética dominio-específicas y ubicuas.

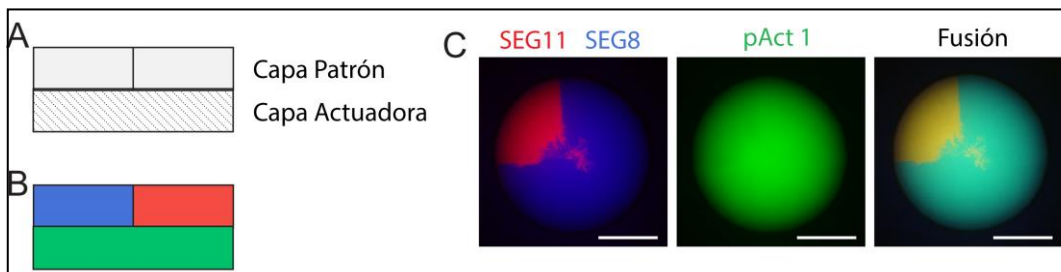


Figura 4-10: Implementación de la capa actuadora. a) Esquema de los niveles de organización en la lógica de capas plasmidiales formadoras de patrón y actuadora. b) Esquema de cómo se implementó la co-localización de funciones genéticas espaciales y ubicuas en dos dominios separados de la colonia. c) Imagen de una colonia bacteriana representativa en la que se prueba este sistema: presencia homogénea de pAct1 (sfGFP) y sectorizada de SEG11 (mCherry) y SEG8 (mTurquoise2). La imagen superpuesta resalta la co-localización de pAct1 y SEG8 o SEG11 en cada dominio. Barra = 500 μ m.

4.3.2 Complementación metabólica dominio específica

Para validar la utilidad de la herramienta diseñada, se instruyeron dominios definidos con funcionalidad metabólica diferenciada dentro de la colonia. Esto se logró mediante la complementación metabólica de los genes incluidos en la capa actuadora ubicua y la capa formadora de patrones. La ruta bio-sintética utilizada fue la de producción del pigmento *violaceína*, la cual corresponde a cinco enzimas provenientes del microorganismo *Chromobacterium violaceum* (August et al., 2000), y que permiten obtener un resultado visible de la complementación.

Dicha ruta metabólica está compuesta por los genes VioA, VioB, VioC, VioD y VioE. Tres de ellos, VioA,B y E, fueron insertados en el plásmido actuador (pAct2), los cuales catalizan la conversión de triptófano en prodeoxiviolaceína, la cual espontáneamente se convierte en un compuesto de color verde llamado *deoxycromoviridians* (Torella,net al., 2014). Tal como se esperaba, este plásmido se expresa de forma ubicua, generando una coloración verde suave en la colonia (Figura 4-11 B).

Los genes VioC y D fueron incluidos en el plásmido segregante A0CD, los cuales no son capaces de generar la producción de ningún pigmento por sí mismos, pero completan la conversión de *deoxycromoviridians* a *violaceína* al ser co-expresados con VioABE. Esto fue comprobado mediante la co-transformación de *E.coli* con A0CD y pAct2, resultado en colonias con la coloración violeta propia del operón completo (Figura 4-11).

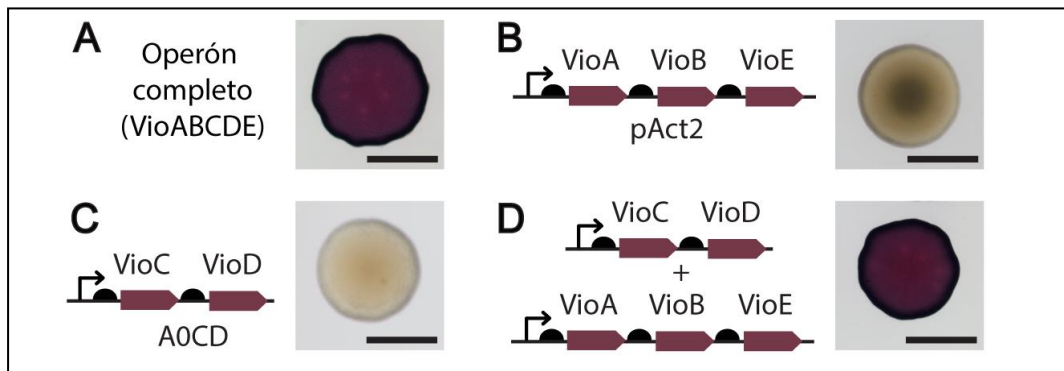


Figura 4-11: Expresión parcial o total de la ruta bio-sintética de violaceína. A) Colonia bacteriana expresando el operón completo de producción de violaceína. B) Esquema de genes insertados en pAct2, los cuales conducen a una coloración verde de la colonia. C) Esquema de los genes insertados en A0CD, los cuales no conducen a una pigmentación de la colonia. D) Colonia bacteriana co-transformada con A0CD y pAct2, que en conjunto incluyen los 5 genes de la ruta metabólica y conllevan la expresión de violaceína. Todas las imágenes corresponden a colonias de *E.coli* representativas. Barra = 500 µm.

Al realizar el ensayo de segregación con SEG9, A0CD y pAct2, se observó expresión ubicua de VioABE con la consecuente pigmentación verde en el dominio con SEG9 y su complementación en la formación de violaceína en el sector de A0CD (Figura 4-12 A). Adicionalmente, se llevó a cabo un control de segregación con SEG9, SEG0C – un plásmido segregante sin color – y pAct2, lo que condujo a la coloración verde de la colonia, pero no violeta en el dominio de SEG0C (Figura 4-12 B). Estos resultados demostraron la capacidad del sistema de dos capas para expresar de rutas bio-sintéticas, de manera dominio específica, mediante la complementación metabólica de dichas capas.

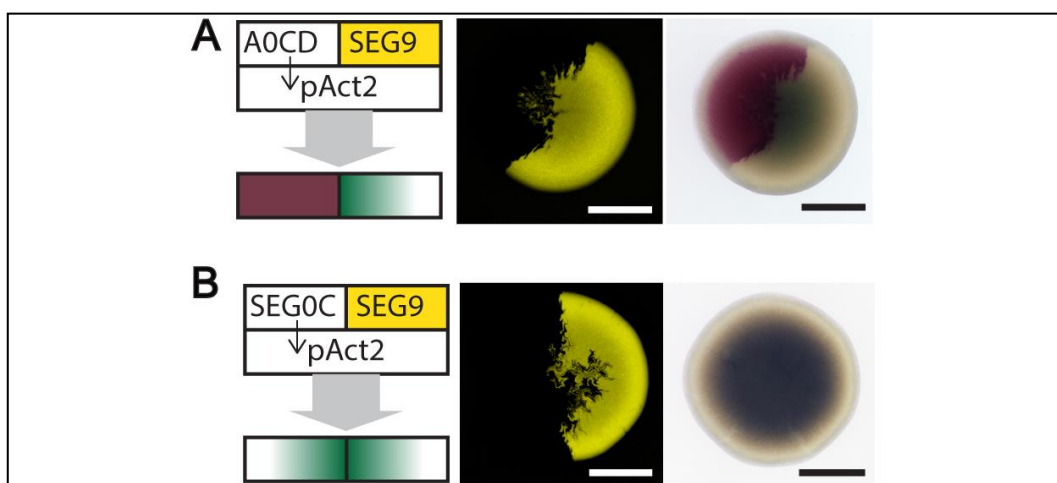


Figura 4-12: Complementación metabólica dominio específica. A) Colonia representativa del ensayo de segregación con SEG9, A0CD y pAct2. B) Colonia representativa del control de segregación con SEG9, SEG0C y pAct2. Ambas sub-figuras incluyen: esquema de las capas y el resultado esperado, imagen obtenida por microscopía de fluorescencia e imagen obtenida por microscopía de campo claro para una colonia representativa en placas de medio M9 suplementado con Cloranfenicol y Kanamicina. Las imágenes fueron tomadas a las 24 h. Barra = 500 μ m.

4.3.3 Control de expresión genética dominio específica con RNAPT7

Para testear la capacidad de regulación *trans* (*i.e.* de un plásmido a otro) del sistema, se ensambló pAct3 el cual contiene mCherry bajo el promotor pT7, el cual es expresado sólo por la polimerasa T7 (Figura 4-13). La RNAPT7 fue incluida bajo el promotor *pLac* en un vector de segregación que fue denominado S8T7 (Figura 4-13). Este promotor fue controlado por la proteína *LacI*, también incluida en el vector, y permitió regular la expresión de RNAPT7 de manera temporal mediante el uso de IPTG (compuesto químico que reprime la acción de *LacI*, activando la expresión del promotor *pLac*).

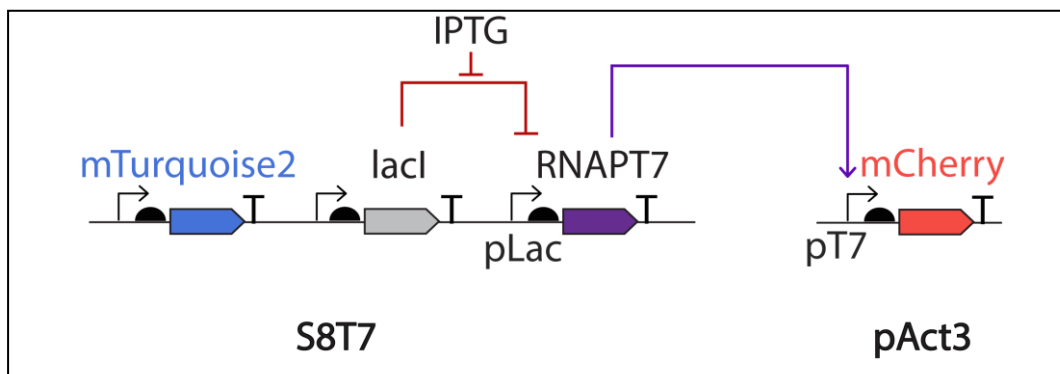


Figura 4-13: Esquema unidades transcripcionales incluidas en S8T7 y pAct3. El plásmido segregante S8T7 comprende 3 unidades transcripcionales: la primera y la segunda expresan de manera constitutiva la proteína fluorescente mTurquoise2 y LacI respectivamente, mientras la tercera expresa la polimerasa T7 bajo el promotor pLac (reprimido por LacI). pAct3 solo contiene una unidad transcripcional capaz de expresar mCherry en presencia de la RNAPT7 (flecha violeta).

En la Figura 4-14 se observa la producción de fluorescencia para cultivos de cepas *E.coli* transformadas con cada plásmido por separado y juntos. Solo se observó expresión de fluorescencia de mCherry en la cepa con los dos plásmidos, la cual mostro un máximo al activar la producción de RNAPT7 con IPTG y un nivel basal en la ausencia de este inductor. Estos resultados demostraron la capacidad de regular de manera *trans*, y con un gatillador externo, la expresión de genes con el sistema de dos capas.

En vista del resultado anterior, se realizaron ensayos de segregación de S8T7 y SEG9 sobre pAct3, ubicuo en la colonia, con el fin de explorar la regulación temporal y espacial de la transcripción. Como producto de este experimento, se obtuvo inducción de la expresión de fluorescencia de manera dominio específica al incluir IPTG en el medio de cultivo (Figura 4-15 A). Por otra parte, no se observó expresión de fluorescencia en el control de segregación con SEG8 y SEG9 sobre pAct3 (Figura 4-15 B).

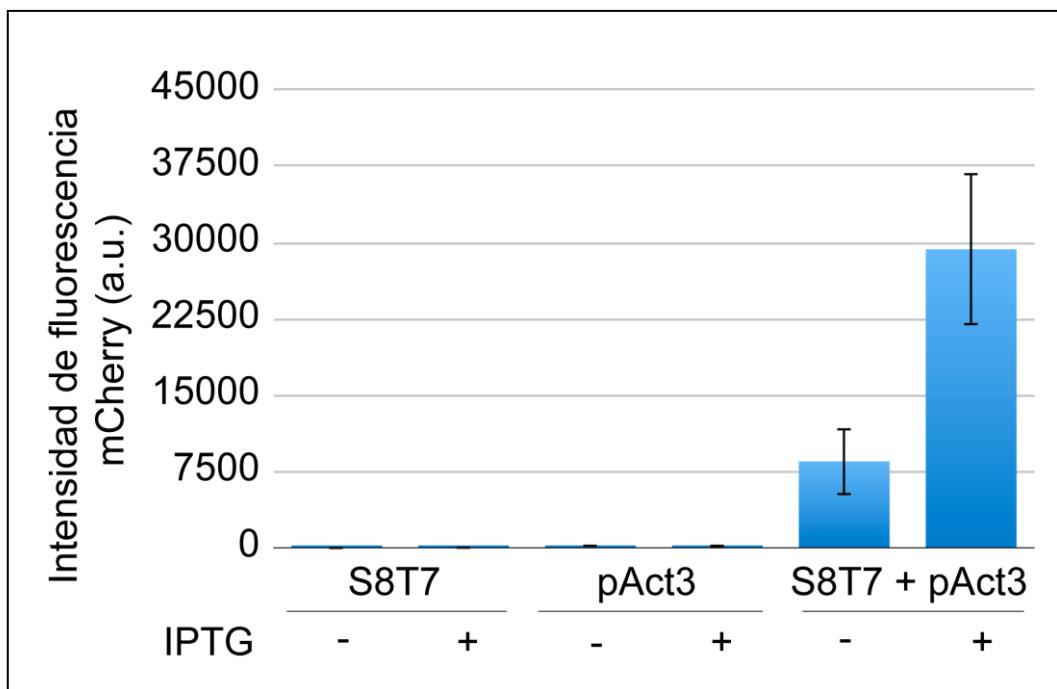


Figura 4-14: Control de la expresión de mCherry con RNAPT7 e IPTG. RNAPT7 se encuentra incluido en S8T7 bajo el promotor pLac, por lo que su expresión solo se activa con IPTG. A su vez la expresión de fluorescencia es activada solo por RNAPT7. No se observa fluorescencia roja en bacterias transformadas con cada plásmido por separado. Al co-transformar se activa la producción de mCherry, con una expresión basal en ausencia de IPTG y un aumento en la señal al incluirlo. Cada columna corresponde al promedio de 10 cultivos celulares y las barras a la desviación estándar muestral. La intensidad de fluorescencia fue medida de acuerdo con lo indicado en sección 3.5. Inducción IPTG = 0,2 mM.

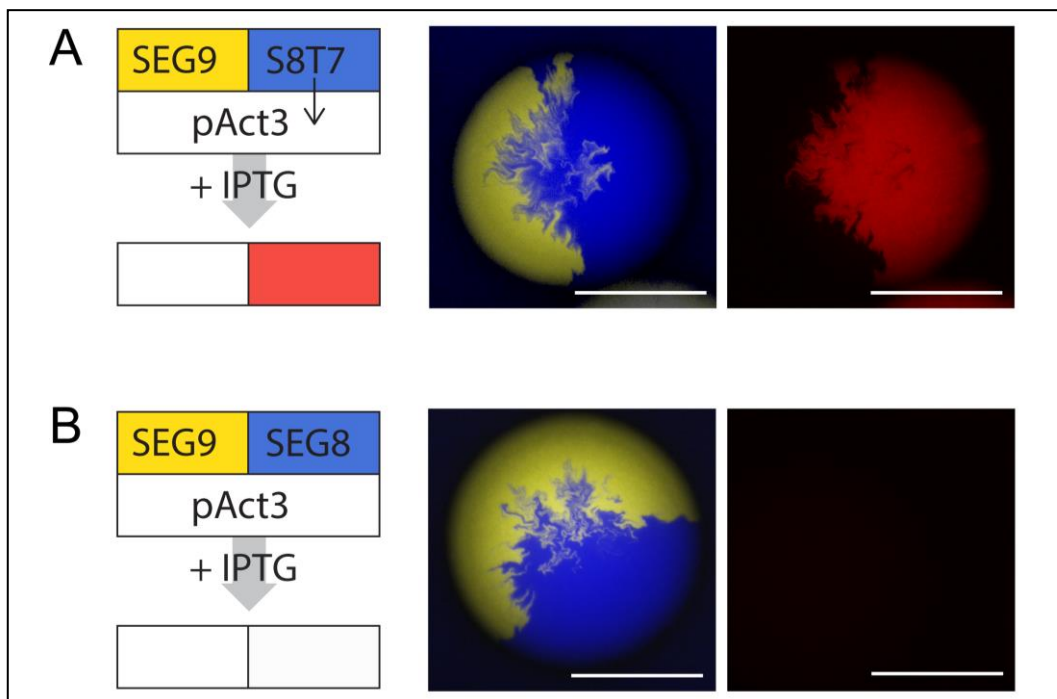


Figura 4-15: Ensayo de regulación genética dominio específica con RNAPT7. Las imágenes corresponden a colonias representativas de cada condición. A) Segregación de SEG9 y S8T7 (capaz de expresar la RNAPT7) que genera inducción de mCherry desde pAct3 de manera dominio específica ante la inducción con IPTG: B) Control de segregación con SEG9 y SEG8 (sin RNAPT7) sobre pAct3 el cual no condujo a la expresión de mCherry ante la inducción con IPTG. Barra = 500 μ m. La intensidad de fluorescencia fue medida de acuerdo con lo indicado en sección 3.5. Inducción IPTG = 0,2 mM.

Finalmente, con el fin de reducir el nivel de expresión basal de mCherry en ausencia de IPTG, se utilizó una nueva versión de RNAPT7 dividida en subunidades (Segall-Shapiro et al., 2014). Estas subunidades son expresadas de manera independiente lo que reduce la probabilidad de que se unan y conformen la RNAPT7 a bajos niveles de expresión. Para esto se construyó S8BST7, que expresaba las subunidades β y σ en tándem bajo el promotor pLac, y pAct4, regulaba la expresión de α bajo el promotor pLac y mVenus bajo pT7 (Figura 4-16 A).

El sistema anterior fue probado directamente en un ensayo de segregación, con SEG11, S8SBT7 y pAct4, obteniendo regulación dominio específica y control temporal sobre el momento de inducción de fluorescencia con IPTG, añadido a las 8 h de crecimiento (Figura 4-16 B). Adicionalmente, se construyó pAct5, con el cual se observó la capacidad de regular dos genes simultáneamente (Figura 4-17). Estos resultados demostraron el uso del sistema de dos capas para la regulación de genes de manera espacial y temporal.

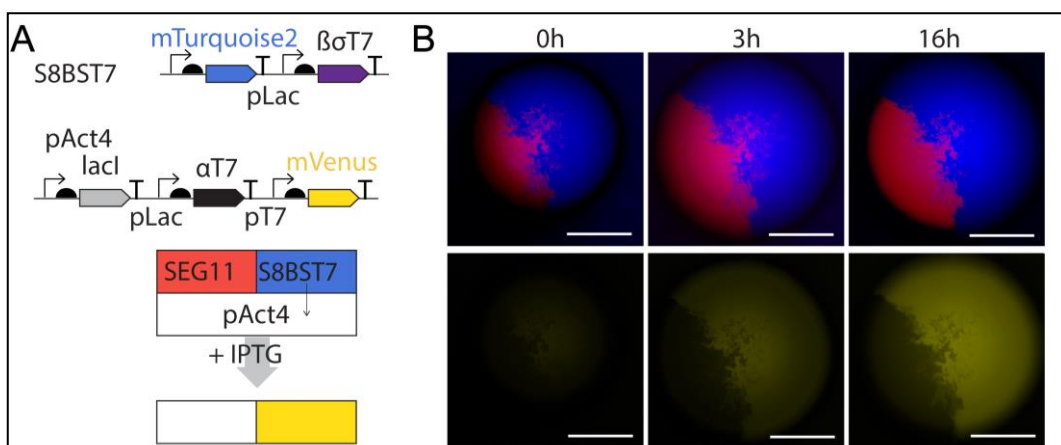


Figura 4-16: Regulación dominio específica y temporal utilizando RNAPT7 dividida. A) Se muestra un esquema de las unidades transcripcionales incluidas en S8BST7, mTurquoise expresado de manera constitutiva y las subunidades β y σ bajo pLac; y en pAct4, LacI constitutivo, la subunidad α bajo pLac y el promotor pT7 controlando a mVenus. El esquema muestra que solo en el dominio donde se expresan las 3 subunidades se espera inducción de fluorescencia al inducir con IPTG. B) Colonia representativa que muestra la capacidad de control sobre el momento temporal de la inducción con IPTG, el cual fue añadido a las 8 h de crecimiento. Barra = 500 μ m. La intensidad de fluorescencia fue medida de acuerdo con lo indicado en sección 3.5. Inducción IPTG = 0,2 mM.

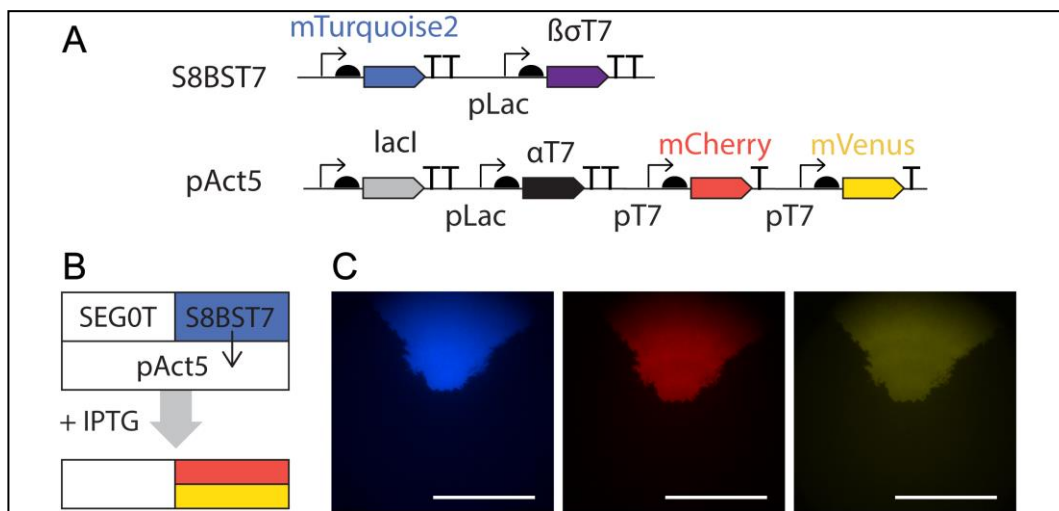


Figura 4-17: Regulación dominio específica de hasta dos genes mediante RNAPT7. A) Se muestra el esquema de las unidades transcripcionales incluidas en plásmidos utilizados para el ensayo, donde mCherry y mVenus depende de la expresión de RNAPT7, la cual a su vez depende de la inducción de IPTG por estar regulada con LacI. B) El esquema del arreglo en dos capas muestra como dicha inducción activa la expresión de fluorescencia roja y amarilla en el dominio de la colonia que contiene a S8BST7. C) Se presenta una colonia representativa del ensayo de segregación e inducción. Barra = 500 μm . Inducción IPTG = 0,2 mM.

4.4 Prototipado de regulación morfogénica

Con el fin de demostrar y explorar el prototipado de funciones morfogénicas a partir de una célula individual, objetivo final de esta tesis, se utilizó el sistema de dos capas SEG# y pAct para instruirlo. Para esto, se propuso la regulación de la tasa de crecimiento celular en dominios específicos como generador de cambios en la forma de la colonia. El sistema utilizado para regular el crecimiento celular correspondió a una aplicación del sistema CRISPRi para regular la síntesis de metionina en *E. coli* (Nuñez et al., 2017), con la cual se gana control sobre la tasa de crecimiento.

Dado que el sistema CRISPRi comprende dos elementos: la proteína dCas9 y el RNA guía (sgRNA), el primero se incluyó de forma ubicua en el plásmido actuador pAct6 bajo el promotor *pLac* (i.e. regulado por IPTG), mientras que el

segundo fue segregado en un dominio específico. Esto último se concretó en los plásmidos S11Ma y S8Ma, que segregan el sgRNA en el dominio rojo y azul, respectivamente, con el fin de descartar un efecto asociado a cada proteína fluorescente y antibiótico (Figura 4-18). De este modo la ruptura de simetría fue realizada con las combinaciones S11Ma – SEG8 y SEG11 - S8Ma.

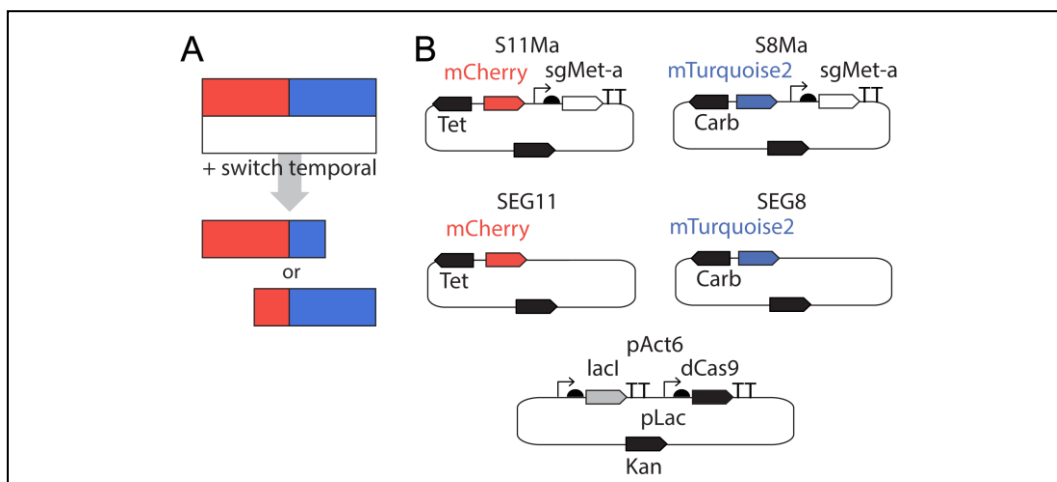


Figura 4-18: Esquema del sistema de inducción de cambios morfológicos. A) Lógica en dos capas de regulación temporal del crecimiento celular de manera dominio específica. B) Esquema de los plásmidos elaborados para llevar a cabo las pruebas de segregación. pAct6 incluye la proteína dCas9 regulada por IPTG, mientras S8Ma y S11Ma comprenden una proteína fluorescente de marcaje y el RNA guía (sgMet-a). SEG8 y SEG11 son los plásmidos control (sin sgMet-a) utilizados.

Para estar seguros de que el efecto en la morfología estaba asociado al sistema genético creado, las pruebas de segregación fueron realizadas en distintas condiciones de medio de cultivo: medio mínimo +/- IPTG y medio mínimo + metionina + IPTG. Este último medio control demostró que la expresión de dCas9 no afectaba el crecimiento de las colonias bacterianas (Figura 4-19 A). Solo las colonias con IPTG y sin metionina mostraron un cambio en la forma (Figura 4-19 A). Los controles respectivos muestran que S8Ma y S11Ma no afectan la forma sin inducción por IPTG, así como los plásmidos control (SEG8 y SEG11) tampoco lo

hacen al añadir el inductor (Figura 4-19 E). Las colonias cuyos sectores no presentaban una continuidad desde el centro hasta el borde o una definición clara de sus dominios, no fueron consideradas para los análisis de forma (Figura 4-19 B).

Para cuantificar el efecto en la morfología, se realizó una caracterización numérica (sección 3.6) de las formas obtenidas en cada caso, tomando como parámetro relevante la excentricidad (Figura 4-19 C). Este descriptor adquiere un valor igual a cero para elementos perfectamente circulares, y aumenta con las desviaciones de esta forma. Para las colonias en los grupos control la excentricidad bordeó un valor de $\sim 0,2$, mientras que en los grupos con inducción de IPTG (y sin metionina añadida) su valor fue casi duplicado. Los análisis estadísticos efectuados sobre los resultados aportan evidencia de una divergencia morfológica clara en estos últimos tratamientos (Figura 4-19 D).

Con el fin de mejorar la comprensión de los efectos generados, se realizaron simulaciones en CellModeller (códigos en anexo C). Para dicho efecto, se creó un módulo en este software capaz de representar el crecimiento celular dependiente del dominio, el que actuó en conjunto al de “segregación” previamente establecido (sección 3.8). Así, fue posible estudiar *in silico* la influencia del patrón inicial (bi-partita o multisector) o el instante de inducción del cambio en la tasa de crecimiento (inducción al momento de tener 0, 1.000, 5.000 o 10.000 células en la colonia). Estas simulaciones presentaron gran similitud con los resultados observados en colonias inducidas con IPTG (Figura 4-20).

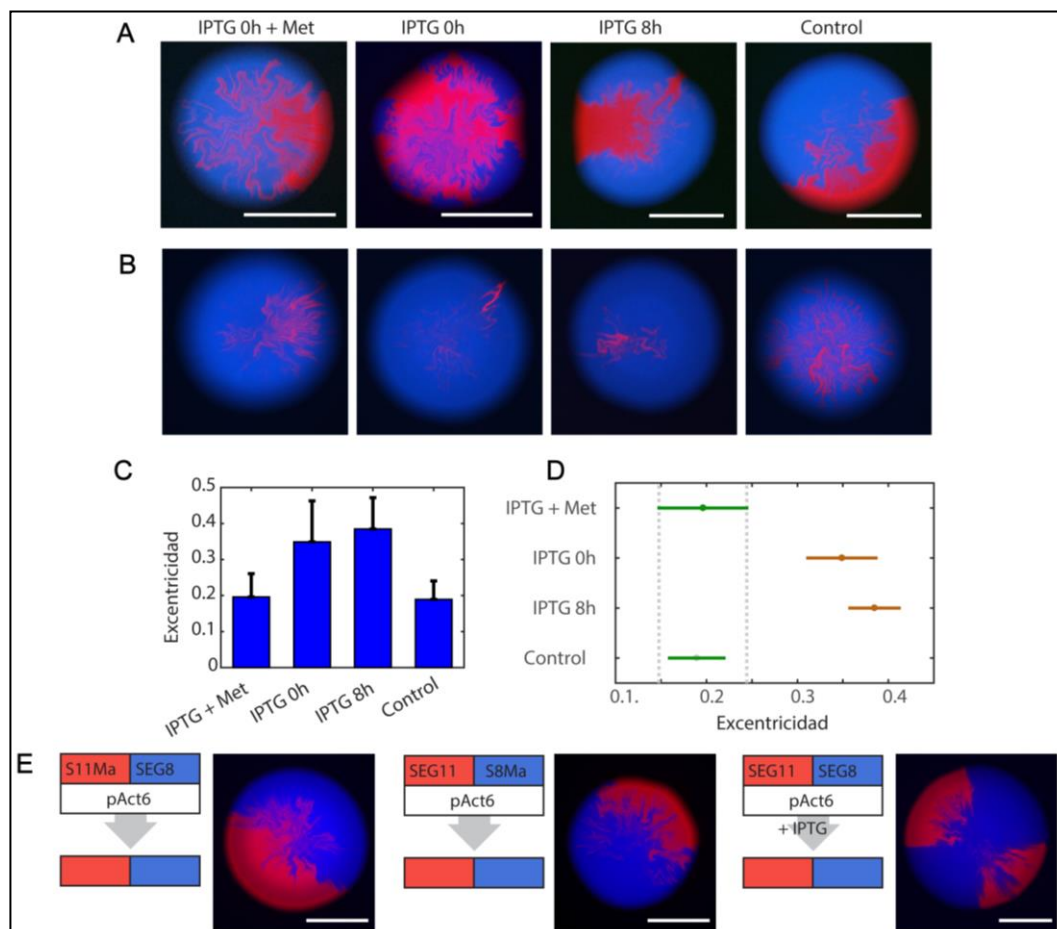


Figura 4-19: Inducción de cambios morfológicos de manera dominio específica en colonias bacterianas. A) Imágenes representativas de colonias bacterianas sometidas a 4 tratamientos distintos: inducción con IPTG a las 0 h + suplementación con metionina (12 mg/L), inducción con IPTG a las 0 h, inducción con IPTG a las 8 h y tratamiento control (sin IPTG ni metionina). Inducción con IPTG = 0.2 mM. Barra = 500 μ m. B) Imágenes representativas de colonias bacterianas descartadas para el análisis por no tener continuidad en el dominio más pequeño (del centro al borde), o no estar bien definido. C) Excentricidad calculada para las colonias de los 4 tratamientos de (A), de acuerdo a este parámetro: círculos $\rightarrow$ $ex = 0$, elipses $\rightarrow 0 < ex < 1$, hipérbolas $\rightarrow ex > 1$. Cada barra es el valor promedio para 15 a 42 colonias medidas, y la barra de error corresponde a la desviación estándar muestral. D) Test multi-comparativo que muestra como los resultados de los 4 tratamientos se separan en dos grupos, en naranja se aprecia que los inducidos a las 0 y 8 horas (sin suplementar con metionina) tienen una diferencia significativa en su media con respecto a los tratamientos IPTG (0 h) + metionina y control (en verde). El cálculo de su ANOVA también apoya la diferencia entre los tratamientos (Prob. F < 6.16e-15, data no presentada). E) Controles para verificar que S11Ma o S8Ma no afectan el crecimiento sin ser inducidos con IPTG, así como que SEG11 y SEG8 tampoco lo hacen al aplicar IPTG. Barra = 500 μ m.

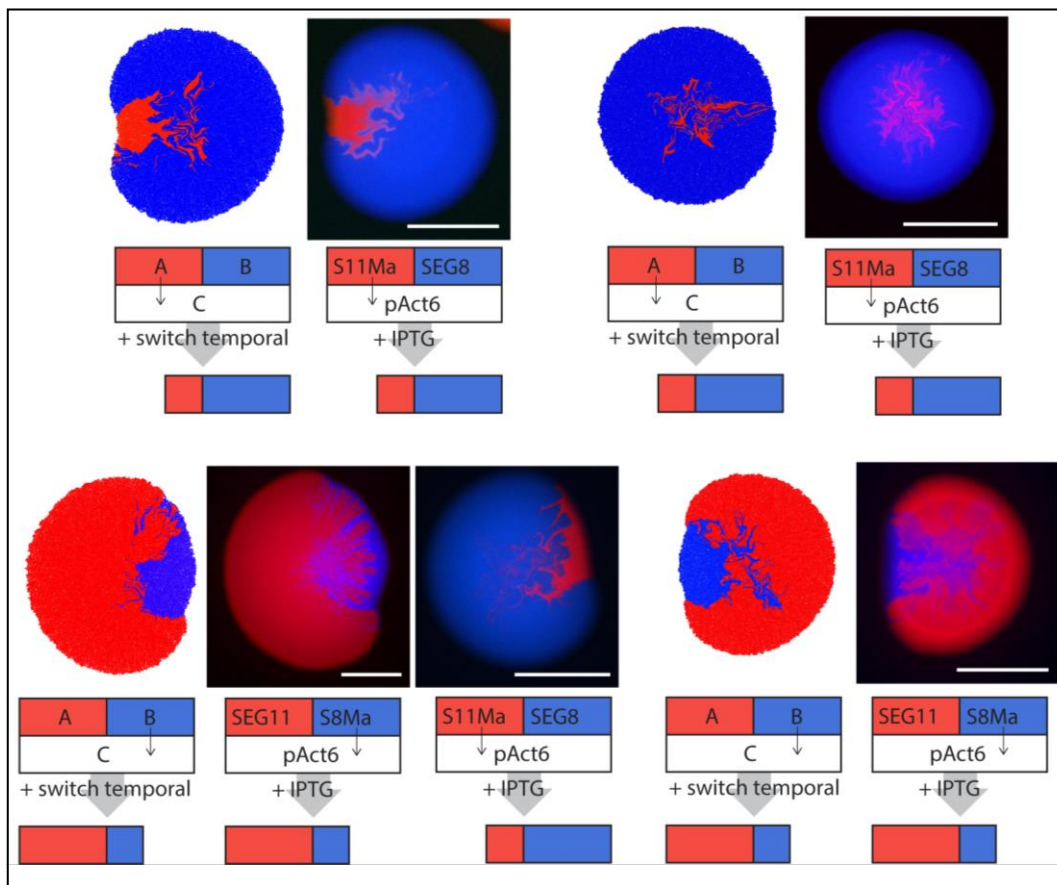


Figura 4-20: Resultados representativos de cambios morfogénéticos. Se presentan imágenes de las morfologías más sobre-representadas, tanto en simulaciones como en data experimental, en tres experimentos independientes. Barra = 500 μ m. La intensidad de fluorescencia fue medida de acuerdo con lo indicado en sección 3.5. Inducción IPTG = 0,2 mM a las 8 h de cultivo y reforzado a las 24 h.

Posteriormente, se realizaron series de tiempo para seguir el efecto del patrón inicial de la colonia en la evolución temporal de su forma (Figura 4-21). Para ello se llevó a cabo un ensayo de segregación con pAct6 y SEG8/S11Ma o S8Ma/SEG11, las cuales fueron inducidas con IPTG a las 8 horas y registradas a las 24 y 48 horas siguientes. En estos experimentos se pudo apreciar la gran similitud de las simulaciones con los resultados experimentales (Figura 4-21 B), siendo CellModeller capaz de capturar las características principales del sistema

real. Esto da cuenta del potencial en el enfoque modular utilizado, principalmente en cuanto a la trazabilidad que imprime en la implementación de instrucciones a colonias bacterianas reales y digitales, así como en la versatilidad de las funciones elementales desarrolladas.

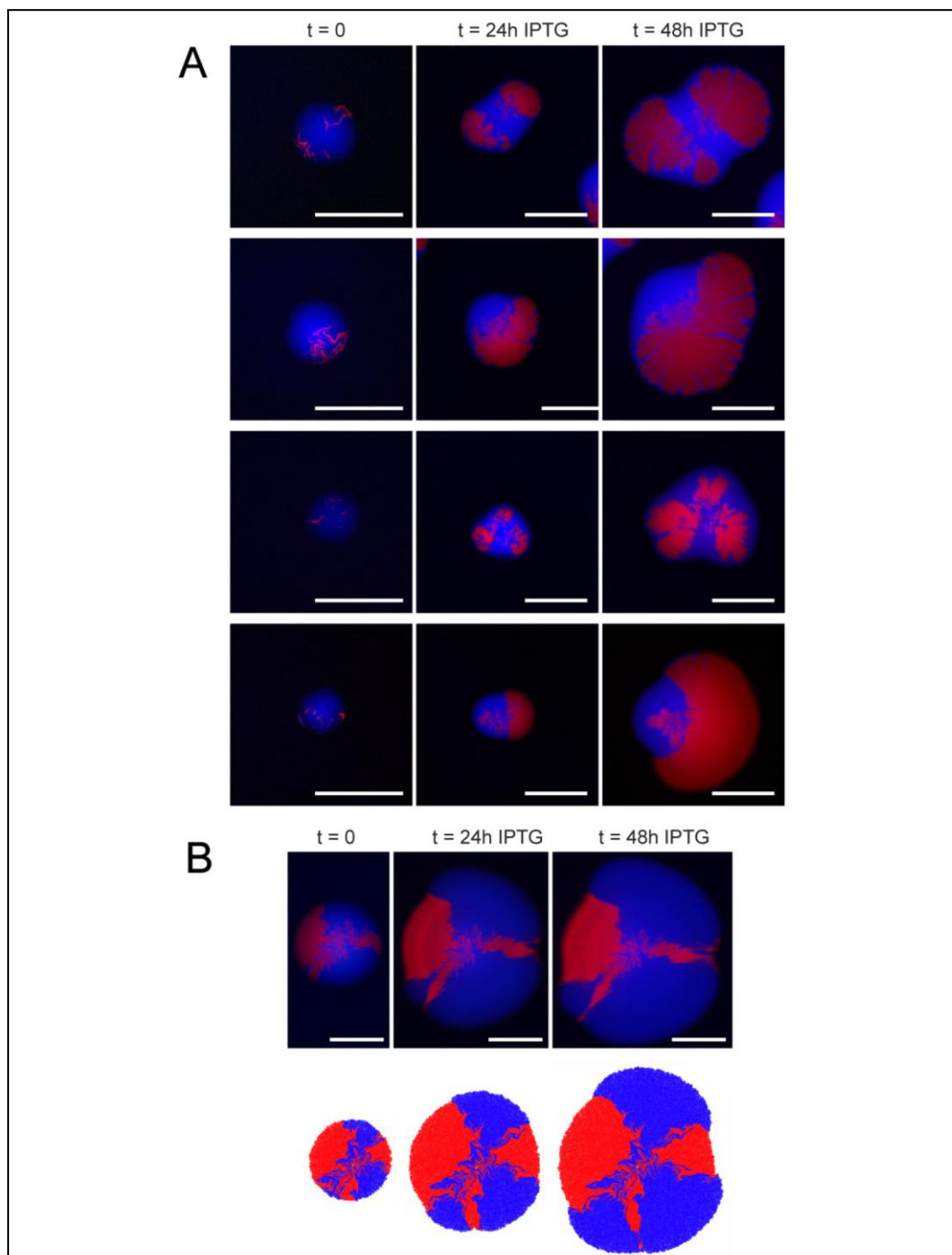


Figura 4-21: Serie de tiempo de efecto morfogenético. Imágenes representativas de cómo el patrón inicial influye en la forma final que adquiere la colonia. A) Series de tiempo para sistema SEG11/S8M. B) Series de tiempo para sistema S11Ma/SEG 8 y su similitud con las simulaciones. La inducción se realizó a las 8 horas ($t=0$) con IPTG 0,2mM, el cual fue reforzado a las 24 horas (0,2 mM). Las imágenes fueron capturadas únicamente en los tiempos indicados en la figura, permaneciendo el resto del periodo en incubadoras a 37°C. Barra = 500 μm .

5 CONCLUSIONES Y PERSPECTIVAS

La ingeniería morfogenética es un área de creciente interés para la programación de la materia viva, capaz de aprovechar sus propiedades de autoorganización y complejidad. Los esfuerzos actuales se han centrado principalmente en su desarrollo teórico (Doursat, Sayama, & Michel, 2013; Goodwin, Kauffman, & Murray, 1993), la creación de sistemas particulares (Liu et al., 2011; Regot et al., 2011) o la formación de patrones (Grant et al., 2016; Payne et al., 2013), faltando herramientas de ingeniería que permitan abordar la generación de formas de manera trazable. Para esto se propuso separar el proceso morfogenético en funciones elementales modulares, como ruptura de simetría y regulación dominio específica. Con ellas se formó una base para el prototipado de mecanismos morfogenéticos, sobre la cual acoplar nuevas funciones elementales que permitan ampliar el espacio de diseño.

En esta tesis, fue posible generar ruptura de simetría en las colonias bacterianas y establecer dominios con límites bien definidos para la expresión genética diferenciada. Este paso resulta fundamental para la generación de polaridad en el sistema y el establecimiento de grupos celulares con funciones o comportamientos resonantes. Con esta base, se diseñó una lógica modular de dos capas, en la cual, la interacción entre los componentes presentes de manera ubicua en la colonia (*e.g.* genoma o plásmido accesorio pAct#) y los componentes distribuidos espacialmente (*e.g.* segregados por vectores SEG#), fue capaz de generar comportamientos controlados espacial y temporalmente. (*e.g.* complementación metabólica y regulación de la expresión genética de manera dominio específica)

Para poder guiar el diseño del sistema y explorar las morfologías posibles, se utilizó el software CellModeller (Rudge et al., 2012), con el cuál se llevaron a cabo simulaciones de los procesos estudiados y se obtuvieron criterios de diseño. Para esto, fue necesario incluir módulos nuevos que permitieron representar las

herramientas genéticas desarrolladas. Particularmente: i) se realizaron análisis estadísticos para definir el número ideal de plásmidos en la célula fundadora que permitirá formar colonias bipartitas. ii) Se exploró el efecto de cambios en el crecimiento celular sobre la morfología de la colonia y el espacio de resultados posibles. Las simulaciones generadas presentaron gran similitud con los resultados obtenidos, lo que sugiere que la plataforma es capaz de capturar las características principales del sistema real. Esto propone que CellModeller puede ser usado como plataforma para el diseño de formas más complejas en proyectos posteriores a esta tesis.

El diseño en dos capas aceleró el proceso de caracterización y facilitó su abstracción en modelos computacionales, dado que permitió manejar por separado actuadores (*i.e.* efecto en la célula) y controladores (*i.e.* generación de patrones y *switch* temporal). Con ello, fue posible integrar un sistema CRISPRi, capaz de regular una vía metabólica esencial para el crecimiento celular, y así instruir cambios en la forma de colonias bacterianas al localizar esta regulación en dominios bien definidos de ella. Esta composición de funciones elementales, y su abstracción computacional, es ventajosa para el prototipado de formas complejas a partir de procesos más simples gestionados en sus unidades fundamentales, lo que permite entender los cambios y deformaciones obtenidas. En vista de esto, es posible verificar la hipótesis planteada, pues el uso de funciones biológicas elementales y abstracciones computacionales permitió obtener trazabilidad en la ingeniería de cambios morfogénéticos en colonias de *E. coli*.

Es importante resaltar la simplicidad y el bajo costo del sistema empleado para este estudio. Las herramientas, metodologías y recursos genéticos concebidos en esta tesis son de libre acceso y fácil de utilizar. Por una parte, se propone el uso de *E.coli* como plataforma de estudio de mecanismos morfogénéticos. Esta bacteria es accesible, fácil de manipular y no posee mayores restricciones de seguridad.

Además, su morfología a nivel poblacional tiene características que siguen leyes mecánicas bien estudiadas, *e.g.* fractales (Rudge et al., 2013), lo que facilita su modelamiento. Este último fue realizado en CellModeller, un software de libre acceso fundamentado en *Python*, que está respaldado por una comunidad de usuarios y puede ser modificado fácilmente según la necesidad del problema estudiado. Finalmente, todas las herramientas genéticas utilizadas están disponibles de manera pública en <https://goo.gl/AbWero>, los protocolos en *protocolos.io* (<https://www.protocols.io/groups/synbiouc-chile>) y el software en *GitHub* (<https://github.com/SynBioUC/MorphoEngineering>).

Cabe destacar que, la toma de imágenes requirió de sofisticados microscopios, lo que aún representa una barrera para el estudio de la morfogenésis. Lo anterior genera una oportunidad para el desarrollo soluciones *open hardware* que aumenten su accesibilidad, la escala de la experimentación reduzca el costo energético y permitan una fácil adaptación a los requerimientos experimentales.

BIBLIOGRAFIA

Adamatzky, A., & Jones, J. (2010). Road planning with slime mould: if *Physarum* built motorways it would route M6/M74 through Newcastle. *International Journal of Bifurcation and Chaos*, 20(10), 3065–3084.

Anderson, P. W. (1972). More is different. *Science*, 177(4047), 393–396.

Arkin, A. P. (2013). A wise consistency : engineering biology for conformity , reliability , predictability. *Current Opinion in Chemical Biology*, 17(6), 893–901. <https://doi.org/10.1016/j.cbpa.2013.09.012>

August, P. R., Grossman, T. H., Minor, C., Draper, M. P., Macneil, I. A., Pemberton, J. M., ... Pharmaceuticals, M. (2000). Sequence Analysis and Functional Characterization of the Violacein Biosynthetic Pathway from *Chromobacterium violaceum*. *Journal of Molecular Microbiology and Biotechnology*, 2(4), 513–519.

Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., & Weiss, R. (2005). A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037), 1130–1134.

Bayer, T. S., Widmaier, D. M., Temme, K., Mirsky, E. A., Santi, D. V., & Voigt, C. A. (2009). Synthesis of methyl halides from biomass using engineered microbes. *Journal of the American Chemical Society*, 131(18), 6508–6515.

Benenson, Y. (2012). Biomolecular computing systems: principles, progress and potential. *Nature Reviews Genetics*, 13(7), 455–468.

Bowman, J. L., Smyth, D. R., & Meyerowitz, E. M. (2012). The ABC model of flower development: then and now. *Development*, 139(22), 4095–4098.

Brophy, J. A. N., & Voigt, C. A. (2014). Focus on Synthetic Biology Principles of genetic circuit design, *11*(5). <https://doi.org/10.1038/nmeth.2926>

Chen, A. Y., Deng, Z., Billings, A. N., Seker, U. O., Lu, M. Y., Citorik, R. J., & Lu, T. K. (2014). Synthesis and patterning of tunable multiscale materials with engineered cells. *Nature Materials*, *13*(5), 515–523.

Church, G. M., Elowitz, M. B., Smolke, C. D., Voigt, C. A., & Weiss, R. (2014). Realizing the potential of synthetic biology. *Nature Reviews Molecular Cell Biology*, *15*(4), 289–294. <https://doi.org/10.1038/nrm3767>

Clancy, K., & Voigt, C. A. (2010). Programming cells: towards an automated “Genetic Compiler.” *Current Opinion in Biotechnology*, *21*(4), 572–581.

Collins, J. J., Gardner, T. S., & Cantor, C. R. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, *403*(6767), 339–342. <https://doi.org/10.1038/35002131>

Del Solar, G., Giraldo, R., Ruiz-Echevarría, M. J., Espinosa, M., & Díaz-Orejas, R. (1998). Replication and control of circular bacterial plasmids. *Microbiology and Molecular Biology Reviews*, *62*(2), 434–464.

Del Vecchio, D. (2015). Modularity, context-dependence, and insulation in engineered biological circuits. *Trends in Biotechnology*, *33*(2), 111–119.

Doursat, R., Sayama, H., & Michel, O. (2012). *Morphogenetic Engineering: Toward Programmable Complex Structures*. Springer, New York. <https://doi.org/10.1017/CBO9781107415324.004>

Doursat, R., Sayama, H., & Michel, O. (2013). A review of morphogenetic engineering. *Natural Computing*, 12(4), 517–535. <https://doi.org/10.1007/s11047-013-9398-1>

Doursat, R., Sayama, H., & Michel, O. (2013). A review of morphogenetic engineering. *Natural Computing*, 12(4), 517–535. <https://doi.org/10.1007/s11047-013-9398-1>

Elhadi, D., Lv, L., Jiang, X.-R., Wu, H., & Chen, G.-Q. (2016). CRISPRi Engineering E. coli for Morphology Diversification. *Metabolic Engineering*, (July). <https://doi.org/10.1016/j.ymben.2016.09.001>

Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767), 335–338.

Elowitz, M., & Lim, W. A. (2010). Build life to understand it. *Nature*, 468(7326), 889.

Endy, D. (2005). Foundations for engineering biology. *Nature*, 438(7067), 449–453. <https://doi.org/10.1038/nature04342>

Fister Jr, I., Yang, X. S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *ELEKTROTEHNISKI VESTNIK*, 80(3), 1–7.

Galanie, S., Thodey, K., Trenchard, I. J., Interrante, M. F., & Smolke, C. D. (2015). Complete biosynthesis of opioids in yeast. *Science*, 349(6252), 1095–1100.

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767), 339–342.

Gibson, D. G., Young, L., Chuang, R., Venter, J. C., Iii, C. A. H., Smith, H. O., & America, N. (2009). Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Methods*, 6(5), 12–17. <https://doi.org/10.1038/NMETH.1318>

Gierer, A., & Meinhardt, H. (1972). A theory of biological pattern formation. *Biological Cybernetics*, 12(1), 30–39.

Goodwin, B. C., Kauffman, S., & Murray, J. D. (1993). Is morphogenesis an intrinsically robust process? *Journal of Theoretical Biology*, 163(1), 135–144.

Grant, P. K., Dalchau, N., Brown, J. R., Federici, F., Rudge, T. J., Yordanov, B., ... Haseloff, J. (2016). Orthogonal intercellular signaling for programmed spatial behavior. *Molecular Systems Biology*, 12(1), 849. <https://doi.org/10.15252/msb.20156590>

Haurwitz, R. E., Jinek, M., Wiedenheft, B., Zhou, K., & Doudna, J. A. (2010). Sequence-and structure-specific RNA processing by a CRISPR endonuclease. *Science*, 329(5997), 1355–1358.

Hoehme, S., & Drasdo, D. (2010). A cell-based simulation software for multi-cellular systems. *Bioinformatics*, 26(20), 2641–2642.

Howard, J., Grill, S. W., & Bois, J. S. (2011). Turing's next steps: the mechanochemical basis of morphogenesis. *Nature Reviews Molecular Cell Biology*, 12(6), 392–398.

Isaacs, F. J., Dwyer, D. J., Ding, C., Pervouchine, D. D., Cantor, C. R., & Collins, J. J. (2004). Engineered riboregulators enable post-transcriptional control of gene expression. *Nature Biotechnology*, 22(7), 841–847.

Jack, B. R., Leonard, S. P., Mishler, D. M., Renda, B. A., Leon, D., & Sua, G. A. (2015). Predicting the Genetic Stability of Engineered DNA Sequences with the EFM Calculator. *ACS Synthetic Biology*, 4(8), 939–943. <https://doi.org/10.1021/acssynbio.5b00068>

Jang, S. S., Oishi, K. T., Egbert, R. G., & Klavins, E. (2012). Specification and Simulation of Synthetic Multicelled Behaviors. *ACS Synthetic Biology*, 1(8), 365–374.

Kosuri, S., & Church, G. M. (2014). Large-scale de novo DNA synthesis: technologies and applications. *Nature Methods*, 11(5), 499–507.

Kreft, J. U., Booth, G., & Wimpenny, J. W. (1998). BacSim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology*, 144(12), 3275–3287.

Larson, M. H., Gilbert, L. A., Wang, X., Lim, W. A., Weissman, J. S., & Qi, L. S. (2013). CRISPR interference (CRISPRi) for sequence-specific control of gene expression. *Nature Protocols*, 8(11), 2180–2196. <https://doi.org/10.1038/nprot.2013.132>

Liu, C., Fu, X., Liu, L., Ren, X., Chau, C. K., Li, S., ... Lenz, P. (2011). Sequential establishment of stripe patterns in an expanding cell population. *Science*, 334(6053), 238–241.

Lou, C., Stanton, B., Chen, Y.-J., Munsky, B., & Voigt, C. a. (2012). Ribozyme-based insulator parts buffer synthetic circuits from genetic context. *Nature Biotechnology*, 30(11), 1137–42. <https://doi.org/10.1038/nbt.2401>

Lu, T. K., Khalil, A. S., & Collins, J. J. (2009). Next-generation synthetic gene networks. *Nature Biotechnology*, 27(12), 1139–1150. <https://doi.org/10.1038/nbt.1591>

Lucks, J. B., Qi, L., Whitaker, W. R., & Arkin, A. P. (2008). Toward scalable parts families for predictable design of biological circuits. *Current Opinion in Microbiology*, 11(6), 567–573.

Makarova, K. S., Haft, D. H., Barrangou, R., Brouns, S. J., Charpentier, E., Horvath, P., ... Van Der Oost, J. (2011). Evolution and classification of the CRISPR–Cas systems. *Nature Reviews Microbiology*, 9(6), 467–477.

Mukherji, S., & van Oudenaarden, A. (2009). Synthetic biology: understanding biological design from synthetic circuits. *Nature Reviews. Genetics*, 10(12), 859–871.

Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., ... Voigt, C. A. (2016). Genetic circuit design automation. *Science*, 352(6281), aac7341. <https://doi.org/10.1126/science.aac7341>

Nielsen, A. A., Segall-Shapiro, T. H., & Voigt, C. A. (2013). Advances in genetic circuit design: novel biochemistries, deep part mining, and precision gene expression. *Current Opinion in Chemical Biology*, 17(6), 878–892.

Nuñez, I. N., Matute, T. F., Del Valle, I. D., Kan, A., Choksi, A., Endy, D., ... Federici, F. (2017). Artificial symmetry-breaking for morphogenetic engineering bacterial colonies. *ACS Synthetic Biology*, 6(2), 256–265.

Olmos, A., & Kingdom, F. A. (2004). A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception*, 33(12), 1463–1473.

Palomares, L. A., & Estrada-Mondaca, S. Ramirez, O. T. (2004). Recombinant Gene Expression: Reviews and Protocols. *Methods Mol Biol*, 267, 15–51.

Pascalie, J., Potier, M., Kowaliw, T., Giavitto, J. L., Michel, O., Spicher, A., & Doursat, R. (2016). Developmental Design of Synthetic Bacterial Architectures by Morphogenetic Engineering. *ACS Synthetic Biology*, 5(8), 842–861. <https://doi.org/10.1021/acssynbio.5b00246>

Payne, S., Li, B., Cao, Y., Schaeffer, D., Ryser, M. D., & You, L. (2013). Temporal control of self-organized pattern formation without morphogen gradients in bacteria. *Molecular Systems Biology*, 9(697), 1–10. <https://doi.org/10.1038/msb.2013.55>

Purnick, P. E. M., & Weiss, R. (2009). The second wave of synthetic biology: from modules to systems. *Nature Reviews Molecular Cell Biology*, 10(6), 410–422. <https://doi.org/10.1038/nrm2698>

Qi, L. S., Larson, M. H., Gilbert, L. A., Doudna, J. A., Weissman, J. S., Arkin, A. P., & Lim, W. A. (2013). Resource Repurposing CRISPR as an RNA-Guided Platform for Sequence-Specific Control of Gene Expression. *CELL*, 152(5), 1173–1183. <https://doi.org/10.1016/j.cell.2013.02.022>

Regot, S., Macia, J., Conde, N., Furukawa, K., Kjellen, J., Peeters, T., ... Sole, R. (2011). Distributed biological computation with multicellular engineered networks. *Nature*, 469(7329), 207–211. <https://doi.org/10.1038/nature09679>

Rhodium, V. A., Segall-Shapiro, T.H., Sharon, B. D., Ghodasara, A., Orlova, E., Tabakh, H., Burkhardt, D. H., ... Voigt, C. A. (2013). Design of orthogonal genetic switches based on a crosstalk map of σ s, anti- σ s, and promoters. *Molecular Systems Biology*, 9(1), 702.

Rudge, T. J., Federici, F., Steiner, P. J., Kan, A., & Haseloff, J. (2013). Cell Polarity-Driven Instability Generates Self-Organized, Fractal Patterning of Cell Layers. *ACS Synthetic Biology*, 2(12), 705–714.

Rudge, T. J., Steiner, P. J., Phillips, A., & Haseloff, J. (2012). Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology*, 1(8), 345–352.

Saka, Y., & Smith, J. C. (2007). A mechanism for the sharp transition of morphogen gradient interpretation in *Xenopus*. *BMC Developmental Biology*, 7(1), 47. <https://doi.org/10.1186/1471-213X-7-47>

Schaerli, Y., Munteanu, A., Cotterell, J., Sharpe, J., & Isalan, M. (2014). A unified design space of synthetic stripe-forming networks. *Nature Communications*, 5(May). <https://doi.org/10.1038/ncomms5905>

Schendel, F. J., Baude, E. J., & Flickinger, M. C. (1989). Determination of protein expression and plasmid copy number from cloned genes in *Escherichia coli* by flow injection analysis using an enzyme indicator vector. *Biotechnology and Bioengineering*, 34(8), 1023–1036.

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7), 671–5.

Schumann, W. (2008). *Escherichia coli* Cloning and Expression Vectors. In *Plasmids: Current Research and Future Trends*, 1-2.

Segall-Shapiro, T. H., Meyer, A. J., Ellington, A. D., Sontag, E. D., & Voigt, C. A. (2014). A “resource allocator” for transcription based on a highly fragmented T7 RNA polymerase. *Molecular Systems Biology*, 10(7), 742.

Siuti, P., Yazbek, J., & Lu, T. K. (2014). Engineering genetic circuits that compute and remember. *Nature Protocols*, 9(6), 1292–300.

Smith, W. P., Davit, Y., Osborne, J. M., Kim, W., Foster, K. R., & Pitt-Francis, J. M. (2017). Cell morphology drives spatial patterning in microbial communities. *Proceedings of the National Academy of Sciences*, 114(3), E280-E286.

Stanton, B. C., Nielsen, A. A., Tamsir, A., Clancy, K., Peterson, T., & Voigt, C. A. (2014). Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nature Chemical Biology*, 10(2), 99–105.

Steinberg, M. S. (1970). Does differential adhesion govern self-assembly processes in histogenesis? Equilibrium configurations and the emergence of a hierarchy among populations of embryonic cells. *Journal of Experimental Zoology Part A: Ecological Genetics and Physiology*, 173(4), 395–433.

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S., & Hasty, J. (2008). A fast , robust and tunable synthetic gene oscillator. *Nature*, 456(7221), 516–519. <https://doi.org/10.1038/nature07389>

Studier, F. W., & Moffatt, B. A. (1986). Use of bacteriophage T7 RNA polymerase to direct selective high-level expression of cloned genes. *Journal of Molecular Biology*, 189(1), 113–130.

Temme, K., Hill, R., Segall-shapiro, T. H., Moser, F., & Voigt, C. A. (2012). Modular control of multiple pathways using engineered orthogonal T7 polymerases. *Nucleic Acids Research*, 40(17), 8773-8781., 40(17), 8773–8781. <https://doi.org/10.1093/nar/gks597>

Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebbber, D. P., Fricker, M. D., ... Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science*, 327(5964), 439–442.

Thomas, C., & Summers, D. (2008). Bacterial Plasmids. *eLS*. <https://doi.org/10.1002/9780470015902.a0000468.pub2>

TOP10 chemically competent cells. (2013). *OpenWetWare*. Recuperado de http://openwetware.org/index.php?title=TOP10_chemically_competent_cells&oldid=677117

Torella, J. P., Boehm, C. R., Lienert, F., Chen, J. H., Way, J. C., & Silver, P. A. (2014). Rapid construction of insulated genetic circuits via synthetic sequence-guided isothermal assembly. *Nucleic Acids Research*, 42(1), 681–689. <https://doi.org/10.1093/nar/gkt860>

Torella, J. P., Lienert, F., Boehm, C. R., Chen, J., Way, J. C., & Silver, P. A. (2014). Unique nucleotide sequence – guided assembly of repetitive DNA parts for synthetic biology applications. *Nature Protocols*, 9, 2075–2089. <https://doi.org/10.1038/nprot.2014.145>

Turing, A. M. (1952). The chemical basis of morphogenesis. *Proc. R. Soc. Lond. B Biol. Sci.*, 237, 37–72.

Velappan, N., Sblattero, D., Chasteen, L., Pavlik, P., & Bradbury, R. M. (2007). Plasmid incompatibility : more compatible than previously thought ? *Protein Engineering Design and Selection*, 20(7), 309–313. <https://doi.org/10.1093/protein/gzm005>

- Wang, B., Kitney, R. I., Joly, N., & Buck, M. (2011). Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology. *Nature Communications*, 2, 508.
- Weber, E., Engler, C., Gruetzner, R., Werner, S., & Marillonnet, S. (2011). A modular cloning system for standardized assembly of multigene constructs. *PLoS ONE*, 6(2). <https://doi.org/10.1371/journal.pone.0016765>
- Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology*, 25(1), 1–47.
- Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R., & Benenson, Y. (2011). Multi-Input RNAi-Based Logic Circuit. *Science*, 333(6047), 1307–1311.
- Young, J. W., Locke, J. C., Altinok, A., Rosenfeld, N., Bacarian, T., Swain, P. S., ... Elowitz, M. B. (2012). Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy. *Nature Protocols*, 7, 80–88.
- Zeng, Y., Li, Y. Y., Chen, J., Yang, G., & Li, Y. (2010). Dendrimers: A Mimic Natural Light-Harvesting System. *Chemistry—An Asian Journal*, 5(5), 992–1005.

ANEXOS

ANEXO A: ARTÍCULO ISI

Artificial Symmetry-Breaking for Morphogenetic Engineering Bacterial Colonies

Isaac N. Nuñez,^{#,†,⊥} Tamara F. Matute,^{#,†,⊥} Ilene D. Del Valle,^{#,‡} Anton Kan,[§] Atri Choksi,^{||} Drew Endy,^{||} Jim Haseloff,[§] Timothy J. Rudge,[†] and Fernan Federici^{*,‡,§,⊥}

[†]Escuela de Ingeniería, Pontificia Universidad Católica de Chile, 7820436, Santiago, Chile

[‡]Departamento de Genética Molecular y Microbiología, Facultad de Ciencias Biológicas, Pontificia Universidad Católica de Chile, 8331150, Santiago, Chile

[§]Department of Plant Sciences, University of Cambridge, Downing Street, Cambridge, CB2 3EA, United Kingdom

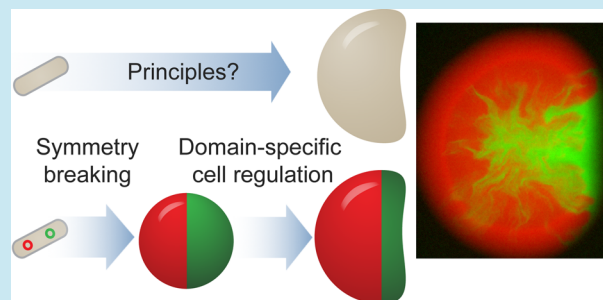
^{||}Department of Bioengineering, Stanford University, Stanford, California 94305, United States,

[⊥]Fondo de Desarrollo de Areas Prioritarias Center for Genome Regulation, Millennium Nucleus Center for Plant Systems and Synthetic Biology, Pontificia Universidad Católica de Chile, 7820436, Santiago, Chile

S Supporting Information

ABSTRACT: Morphogenetic engineering is an emerging field that explores the design and implementation of self-organized patterns, morphologies, and architectures in systems composed of multiple agents such as cells and swarm robots. Synthetic biology, on the other hand, aims to develop tools and formalisms that increase reproducibility, tractability, and efficiency in the engineering of biological systems. We seek to apply synthetic biology approaches to the engineering of morphologies in multicellular systems. Here, we describe the engineering of two mechanisms, symmetry-breaking and domain-specific cell regulation, as elementary functions for the prototyping of morphogenetic instructions in bacterial colonies. The former represents an artificial patterning mechanism based on plasmid segregation while the latter plays the role of artificial cell differentiation by spatial colocalization of ubiquitous and segregated components. This separation of patterning from actuation facilitates the design-build-test-improve engineering cycle. We created computational modules for CellModeller representing these basic functions and used it to guide the design process and explore the design space *in silico*. We applied these tools to encode spatially structured functions such as metabolic complementation, RNAPT7 gene expression, and CRISPRi/Cas9 regulation. Finally, as a proof of concept, we used CRISPRi/Cas technology to regulate cell growth by controlling methionine synthesis. These mechanisms start from single cells enabling the study of morphogenetic principles and the engineering of novel population scale structures from the bottom up.

KEYWORDS: morphogenetic engineering, synthetic biology, morphogenesis, CRISPR, modeling



Synthetic biology is changing our perception of biological systems as a mere source of raw material to a vision of biology as a programmable substrate for material fabrication, chemical production, and computing.^{1–11} The use of design specifications, computer-assisted mathematical modeling, and novel DNA fabrication methods have made possible the engineering of biological functions and systems of increasing reliability, scale and complexity.^{7,8,12–18} However, the scalability of circuit size within single cells has been limited by metabolic burden effects, cross-talk interactions, and genetic instability, a limitation reflected in the number of engineered components per cell.¹⁹ Engineering at the multicellular scale has emerged as a solution to the limitations encountered at the single cell level. The engineering of multicellular functions has been successfully used to engineer artificial consortia, patterning, and distributed computing in microbial systems.^{6,20–27} Defining elementary functions for morphogenetic engineering could complement

these developments by providing mechanisms for the establishment and maintenance of higher order structures. Morphogenetic engineering, is an emergent field founded by Doursat et al.^{28,29} that explores the artificial design of autonomous systems capable of developing complex morphologies. Its emphasis is on the programmability and controllability of self-organization toward the reproducible emergence of collective architectures. Morphogenetic engineering will be critical for engineering tissues, microbial consortia, and living functional materials.³⁰

Engineering morphogenetic mechanisms in biological systems is a challenge that requires the formulation of new approaches and tools. Morphogenesis involves multiparallel, collective and emergent processes across different spatial and temporal scales,^{31,32} which makes the process intractable from

Received: May 21, 2016

Published: October 30, 2016



the engineering perspective. Engineering practices rely on design specifications and elementary functions that are easy to abstract, simulate and implement with predictable core components. This provides tractability for the design-build-test iterative cycle, making the process more informative and reproducible. Aircraft engineering, for instance, relies on principles of flight: lift, propulsion and control; whereas birds integrate these three elementary functions as a whole in their flapping wings (reviewed in^{33,34}). Although the use of fixed wings and the decoupling of lift, propulsion and control seem unnatural and almost contradictory from the avian flight perspective, it provides the mechanistic abstractions that makes the engineering process more tractable. This approach is in line with the synthetic biology mantra of abstraction, modular design and principle-based engineering that allows implementation from characterized core components that fit into the design specifications of the system. Abstracting the complexity of biological processes into well-defined elementary functions has already facilitated the engineering of multicellular patterns.^{21,23,25,35,36} Computational methods for *in silico* engineering of shapes from fundamental collective phenomena have been developed.³⁸ However, the development of tools and resources for prototyping morphogenetic mechanisms remains less explored.^{29,39,40}

Here, we describe the development of computational models and genetic tools for prototyping morphogenetic mechanisms in bacterial colonies. With the aim of gaining tractability, debugging capabilities, and scalability in the process, we have defined two elementary functions: symmetry-breaking and domain-specific cell regulation. These are inspired by two natural process: (i) symmetry-breaking in developing embryos and (ii) domain-specific organogenesis during flower development, a process guided by the colocalization of transcriptional regulators.⁴¹ We implemented these two functions as a system of ubiquitous and segregating plasmids that facilitates modular and combinatorial prototyping of instructions. We also defined these elementary functions as explicit computational modules in CellModeller,^{42,43} which helped to define important properties of the system. CellModeller is an individual-based biofilm modeling platform that computes the growth of colonies of rod-shaped bacteria from a single initial cell. It uses a growing rigid-body method to compute the non-overlapping positions and orientations of elongating cells subject to minimizing work done in overcoming viscous drag. This leads to dominant growth at the colony margin. It also allows simulation of internal cell states such as plasmid content and effects of plasmid gene expression. We applied these methods and resources to the programming of bacterial colony shapes using CRISPRi/Cas9 regulation of cell growth.

■ RESULTS AND DISCUSSION

Overview of the Platform: A Two-Step Mechanism for Morphogenetic Programming. The goal of this work was to develop a simple system for prototyping morphogenetic engineering mechanisms. This system would allow starting from a single cell and controlling shape at the multicellular level using growing bacterial colonies as simple systems (Figure 1A). With the aim of having more tractability during the design-build-test process, we defined two abstract elementary functions: (i) symmetry-breaking and (ii) domain-specific cell regulation (Figure 1B). The first step was intended to establish polarity and domains of cells in the colony; while the second was intended to regulate cells differentially within these

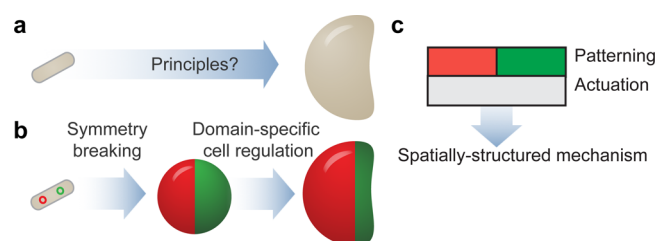


Figure 1. Abstract elementary functions for morphogenetic programming. (a) Schematic representation of a morphogenetic engineering challenge. (b) Adoption of two elementary functions to address this task: symmetry-breaking and domain-specific cell regulation. (c) Schematic representation of the two-tier organization of genetic layers that implement these elementary functions.

domains (Figure 1B). To implement these two steps, we sketched a two-tier organization model inspired by the ABC model of flower development that exploits the spatial coexistence of transcriptional regulators to instruct organogenesis⁴¹ (Figure 1C). This design would allow the implementation of instructions by interchanging plasmid combinations.

Patterning Layer: Symmetry Breaking Function for the Establishment of up to Four Different Domains Per Colony. Symmetry-breaking phenomena, as the process of reducing homogeneity to generate more structured systems,⁴⁴ have been intensively studied in chemical, physical, and biological systems.^{32,45–50} From an engineering point of view, symmetry breaking is key in establishing the first coordinates within an otherwise homogeneous system from where higher order structures and dynamics can be built in search for increasing functional specialization at different scales.

We implemented a symmetry-breaking step, the first elementary function, as a system of segregating plasmids (Figure 2A). This mechanism is based on an artificial tool created to label multiple cell lineages in growing bacterial colonies.⁴³ These plasmids create domains of cells in colonies by segregating from a single cell acting as a colony founder (Figure 2B). This process relies on the segregation of two plasmid variants containing different antibiotic resistance genes (e.g., A1 or A2), spectrally distinct fluorescent proteins but a common plasmid backbone with a third antibiotic resistance gene (e.g., A3) (Figure 2B). When cotransformed and grown in the presence of antibiotics A1 and A2, both plasmid variants are maintained in the same cell. When transferred to a growth media containing antibiotic A3 only, cells are unable to distinguish between the two plasmid variants, leading to a stochastic segregation of these two into different daughter cells. This creates cell lineages harboring only one of the two plasmid variants and its corresponding fluorescent protein (Figure 2C). This system relies on the mechanical properties of bacterial colony growth⁴³ to create radial domains of cells as colonies grow from a single cell. From a morphogenetic engineering perspective, we were interested in the establishment of two to six domains per colony. Colonies composed of two domains only (i.e., bipartite) were of particular interest due to their simplicity and resemblance to the process of polarity establishment in developing embryos.⁵⁰

We created biophysical models of these processes to guide the engineering of these patterned colonies (Figure 2C) (see Supporting Information Movie S1, S2, and S3). We designed a CellModeller plasmid segregation module to explore the effects of copy number on the formation of sectors within colonies.

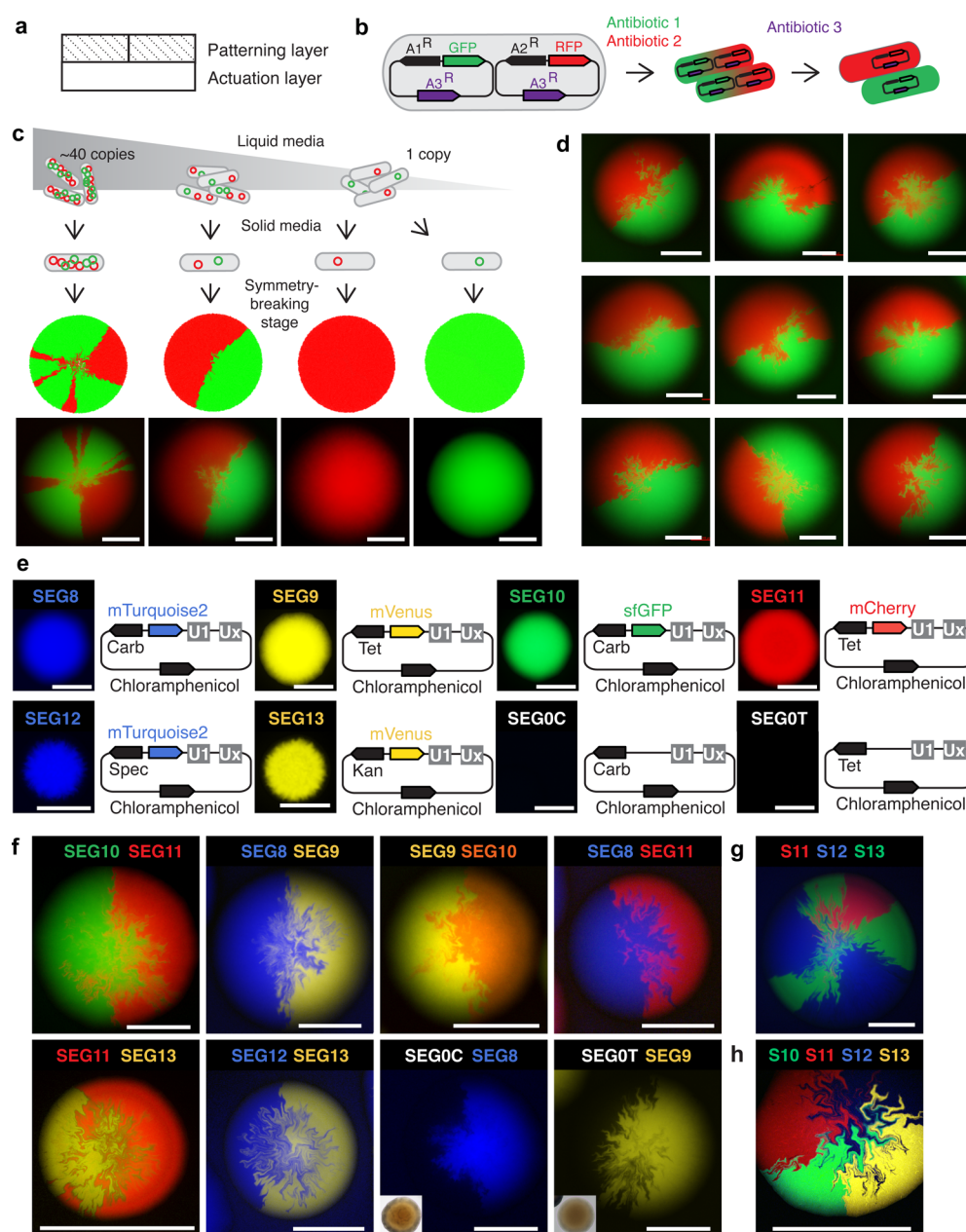


Figure 2. Genetic implementation of the patterning layer. (a) Schematic representation of the two-tier plasmid organization highlighting the patterning layer. (b) Schematic representation of the antibiotic-based regulation of plasmid maintenance within cells. Growing cells in the presence of A1 and A2 antibiotics maintains the two plasmid variants (red and green) within cells. These variants segregate in the presence of antibiotic A3 cells due to the presence of an antibiotic resistance gene located in the backbone that is common to both plasmid variants. Thus, cells maintain plasmids without any preferences for the two variants (either red or green) giving rise to cells containing one or the other (shown as red and green cells). (c) Schematic representation of copy number effect on the formation of bipartite and multisector colonies. The protocol starts with cells growing in liquid media containing arabinose, the high copy number inducer. Plating cells (i.e. colony forming units) at different points after removing arabinose from liquid culture permits starting colonies with different amounts of the two plasmid variants. Plating at later stages, when cells are close to maintain only two copies/cell of the plasmid, increases the likelihood of creating bipartite colonies as shown by CellModeller simulations and experimental data (bottom). (d) Representative bipartite colonies obtained with this protocol. (e) Epifluorescence image of colonies bearing different variants of SEG plasmids: SEG8 (blue), SEG9 (yellow), SEG10 (green), SEG11 (red), SEG12 (blue), SEG13 (yellow), and SEG0C/SEG0T, which do not carry a fluorescent marker in their backbone. (e–g) Colonies showing the segregation of different combinations of two (f), three (g), and four (h) SEG plasmids. Scale bar, 500 μm.

Biophysical parameters of growth were chosen based on a previous work to reproduce the growth of colonies in high resolution confocal microscopy. Growth was restricted to 2-dimensions for simplicity. Simulations were initiated with a single cell containing equal numbers (N) of each of two types of plasmid ($2N$ plasmids in total). On cell division each plasmid

replicated and each daughter cell inherited $2N$ randomly selected plasmids with equal probability of each type. Simulations starting with cells containing 10 (see [Supporting Information Figure S1A](#)) and 4 copies (see [Supporting Information Figure S1B](#)) only produced colonies with multiple domains. To test this further, we engineered segregation

plasmids bearing the low copy origin of replication pSC101*,¹⁶ considered to maintain 4–10 copies per cell. As suggested by simulations, these plasmids created colonies of multiple domains and failed to show the formation of bipartite colonies (see Supporting Information Figure S1C).

According to our model, we needed to reduce the copy number to an ideal state of 2 copies per cell (one copy of each of the two plasmid variants) in order to favor the formation of bipartite colonies with a possibility of 33% (see Supporting Information Figure S1D–F). We used a backbone from the pDestBAC plasmid,⁵¹ whose copy number can be controlled with arabinose from 1 to ~40 (see Supporting Information Figure S2A–C). This system allowed us to maintain the two plasmids in the same cell at medium copy while grown in liquid culture supplemented with A1, A2, and arabinose (i.e., “plasmid propagation stage”) and decrease to single copy when transferred to liquid culture without the arabinose (i.e., “plasmid copy decay stage”) (Figure 2C). Thus, we could grow single cells as colony-forming units in solid media at the point when they contained 2 copies per cell, favoring the formation of bipartite colonies (i.e., “symmetry-breaking stage”).

We used tetracycline and ampicillin resistance cassettes for A1–A2 coselection during the plasmid propagation stage; and a chloramphenicol resistance marker in the backbone as the common antibiotic resistance for the plasmid copy decay stage and the symmetry-breaking stage (Figure 2C, see Supporting Information Figure S2A). We developed a protocol to increase the probability of generating bipartite colonies by changing the length of time that cells were maintained in the plasmid decay stage before being plated onto solid media for symmetry-breaking (Figure 2C). Two independent experiments indicated that 180 min of growth at the plasmid decay stage produced the maximum ratio of bipartite colonies (see Supporting Information Figure S2D–E). Following this protocol, we were able to create bipartite colonies reliably (Figure 2C, see Supporting Information Figure S2F), demonstrating the implementation of the symmetry-breaking elementary function. Unexpectedly, we also observed bipartite colonies at earlier stages when cells were supposed to be in high copy state (see Supporting Information Figure S2F). This could be due to the variability on copy number per cell and unbalanced proportion of plasmid variants in cells grown in liquid cultures. This will inevitably cause a wider distribution of states at the moment of plating cells on solid media.

Being able to program a single cell (colony founder) to trigger morphogenetic changes taking place at the colony level was essential for our aim of prototyping morphologies from the bottom up. It was therefore essential to avoid false bipartite colonies, products of colonies “crashing” into each other during growth in close proximity. For this, we performed controls to identify cell dilutions that increase the chances of getting those unwanted colonies. We found that having a density above 7 colonies/cm² always produced false colonies. We therefore work with densities below 1–3 colonies/cm², which we control by diluting cells accordingly before plating (see Supporting Information Figure S3).

Next, we created a series of SEG# plasmids carrying different fluorescent protein genes in their backbone (sfGFP, mTurquoise2, mVenus, and mCherry) and no fluorescent markers (Figure 2D). The use of four different antibiotic resistance cassettes (kanamycin, spectinomycin, tetracycline, and ampicillin) allowed us to segregate different combinations of two, three, and up to four plasmids (Figure 2E–G). These plasmids

showed no significant effect on cell growth (see Supporting Information Figure S4). These vectors were also engineered to contain unique nucleotide sequences “UNS”⁴⁹ that facilitate combinatorial assembly of instructions from a library of transcriptional units. We created a library of UNS-flanked transcriptional units containing terminators from a list of synthetic strong terminators⁵² (see Supporting Information Figure S5). These results demonstrate that we could use this system to control symmetry-breaking of growing bacterial colonies and establish up to four different cellular states reliably.

Actuation Layer: An Ubiquitously Distributed Plasmid to Encode Domain-Responsive Functions. Following the organizational strategies describe in Figure 1, we created vectors of ubiquitous localization across the colony (Figure 3).

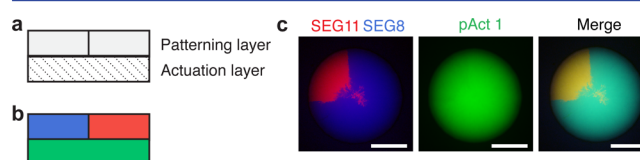


Figure 3. Genetic implementation of the actuation layer. (a) Schematic representation of the two-tier plasmid organization highlighting the actuation layer. (b) Schematic representation of the genetic implementation by the colocalization of the patterning layer composed of SEG vectors (in blue and red), and actuator layer composed of pAct ubiquitous vector (in green). (c) Genetic implementation of this organization by SEG plasmids SEG8 and SEG11, and actuation plasmid pAct1 expressing mVenus (in green). A merged image is shown to highlight the colocalization of plasmids. Scale bar, 500 μm .

These pAct vectors contained actuator genes that regulate different cellular processes in response to or in combination with other regulators being expressed from the patterning layer (SEG plasmids). This process corresponds to the second step in our approach, the domain-specific regulation of cellular processes (Figure 1B). pAct# vectors were designed as low copy plasmids (pSC101*,¹⁶) containing UNS sites to facilitate the combinatorial assembly of transcriptional units. To demonstrate the ubiquitous and sectorized localization of genes from pAct and SEG plasmids, respectively, we expressed mVenus from pAct1, mTurquoise2 from SEG8, and mCherry from SEG11 (Figure 3B,C). These results show how this two-tier organization can be used to superimpose spatial rules over ubiquitous functions.

Domain-Specific Regulation of Colonies: Metabolic Complementation, RNAP7 Activation and CRISPR-Regulation of Gene Expression. To validate the utility of this two-tier organization, we used it to control a series of different cellular functions within the created domains. The concept of domains, along with the maintenance of their boundaries, has been central to the understanding of biological morphogenesis (see a review, ref 53). We propose the use of these domains to assign artificial states and functions to a different part of the colony. First, we applied it to the domain-specific complementation of an incomplete metabolic pathway. We used the five-enzyme metabolic pathway from *Chromobacterium violaceum* controlling the production of violacein.⁵⁴ A synthetic operon containing the five enzymes VioA, VioB, VioC, VioD, and VioE, was used in these experiments (Figure 4A). We constructed pAct2 carrying genes vioA, B, and E that catalyze the conversion of tryptophan to prodeoxyviolacein, which spontaneously become deoxychromoviridans, a green-

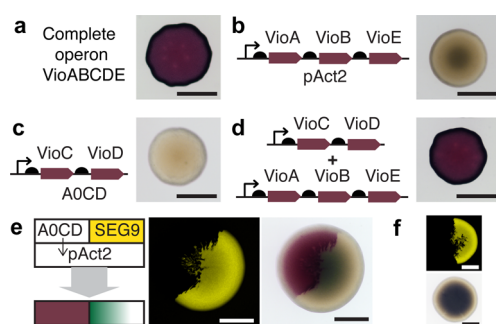


Figure 4. Domain-specific metabolic pathway complementation. (a) Colony showing violacein production from complete VioABCDE violacein operon. (b) Colony expressing *vioC* and *vioD* genes from vector A0CD. (c) Colony showing deoxychromoviridans production from vector pAct2 containing genes *vioA*, *vioB*, and *vioE*. (d) Colony showing violacein production from the cotransformation of vectors A0CD and pAct2. (e) Domain-specific metabolic complementation of violacein synthesis pathway. SEG9 was used as a segregation partner of A0CD. (f) Segregating control SEG9 and SEG0C on top of pAct2 showed no violacein synthesis. Scale bar, 500 μm .

colored compound⁵¹ (Figure 4B). This plasmid was expressed ubiquitously across the colony producing a faint green color in cells located in the center of the colony (Figure 4B). Next, we constructed the segregating plasmid A0CD carrying the genes *vioC* and *vioD*, which was unable to produce any visible pigment on its own (Figure 4C) but produced a purple pigment in combination with *vioABE* due to the conversion of prodeoxyviolacein to violacein (Figure 4D). When segregated with SEG9 on top of ubiquitously located pAct2, A0CD produced violacein in a domain-specific manner as expected from the complementation of ABE pathway with CD genes (Figure 4E). No violacein synthesis was observed when SEG9 and pAct2 were combined with control SEG0C (Figure 4F), a colorless segregating plasmid (Figure 4D).

Next, we explored the segregation of transcriptional controllers acting on target genes located on pAct vectors. We created S8T7 segregation plasmid, carrying the RNA polymerase from phage T7, and pAct3, carrying *mCherry* gene downstream of T7 promoter. The T7 RNAP was expressed from a pLac promoter in order to regulate its expression with a *lacI* repressor constitutively expressed from pAct3. We used IPTG to regulate the expression of *mCherry* from pAct3 in the presence of S8T7 (see Supporting Information Figure S6A). The segregation of S8T7 and SEG9 plasmids on top of pAct3, located ubiquitously across the colony, showed domain-specific activation of *mCherry* upon IPTG induction (Figure 5A). Conversely, control experiment segregating SEG9 and SEG8

failed to induce *mCherry* from pAct3 (see Supporting Information Figure S6B). Next, we used split T7 RNAP⁵⁵ to reduce background expression in the absence of IPTG. We created a segregating plasmid S8BST7 and ubiquitous plasmid pAct4 carrying the $\beta\sigma$ and α fragment of T7 RNAP, respectively (Figure 5B). The segregation of S8BST7 with SEG11 led to domain-specific spatial regulation while IPTG provided temporal control of *mVenus* expression (Figure 5B). This system was shown to activate two target genes simultaneously (see Supporting Information Figure S6C).

To further explore the utility of our system, we performed CRISPRi/Cas9 regulation of gene expression in a domain-specific manner. We targeted VioABCDE synthetic operon to repress the production of the visible pigment violacein. First, we made constructs to test different sgRNAs targeting *VioA* and *VioC* genes (see Supporting Information Figure S7), and selected sgRNA-*VioC*. Next, we created the segregation vector SVioC containing constitutively expressed sgRNA-*VioC* and dCas9. This construct shut down violacein production in a domain-specific manner when segregated with SEG0C on top of the ubiquitously expressed VioABCDE operon (Figure 6A). All these experiments together demonstrated the versatility of our system for the spatial and temporal regulation of cellular functions in bacterial colonies.

Prototyping Morphogenetic Instructions by Symmetry-Breaking and Domain-Specific Cell Growth Regulation. We applied this toolkit to the creation of a tractable morphogenetic mechanism in bacterial colonies. We used CRISPRi/Cas9 regulation of *MetA* gene expression as a mechanism to regulate colony growth via induced methionine auxotrophic cells. First, we designed a series of vectors containing a *LacI*-regulated dCas9 and the constitutive expression of one of four sgRNAs (sgMet-a, -b, -c, -d) targeting the *metA* gene in the genome. These plasmids also included a 5' fusion of the first 156 nt of *MetA* to sfGFP as a fluorescent reporter of *MetA* expression inhibition (see Supporting Information Figure S8A). After comparing the inhibition strength of the four sgRNAs (see Supporting Information Figure S8B), sgMet-a was selected for CRISPRi-induced cell growth regulation (see Supporting Information Figure S8C and D). We added methionine to demonstrate that the effects were due to *MetA* regulation and not by dCas9 expression (see Supporting Information Figure S8D). Following the two-tier approach, we created S11Ma and S8Ma vectors to segregate sgMet-a as red and blue domains, respectively, and pAct6 to ubiquitously express dCas9 under IPTG regulation (Figure 7A and B). This system was proven successful in the regulation of cell growth in liquid culture (see Supporting Information

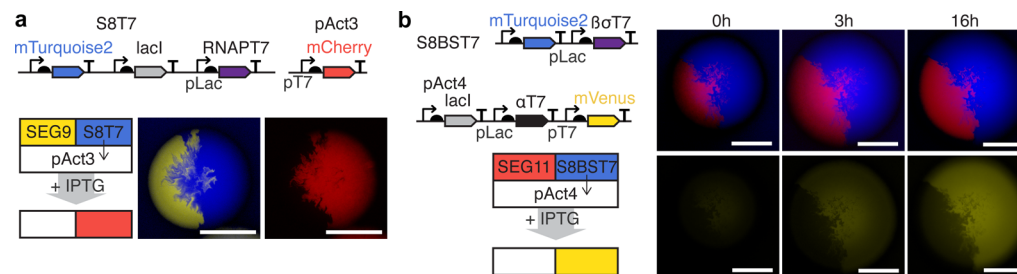


Figure 5. Domain-specific activation of gene expression. (a) Segregation of RNAP T7 from S8T7 (in combination with SEG9) showing the induction of *mCherry* from pAct3 upon IPTG treatment. (b) Temporal and spatial regulation of *mVenus* induction in colonies segregating SEG11 and S8BST7 in combination with pAct4. IPTG induced α and $\beta\sigma$ RNAP T7 fragments from pAct4 and S8BST7, respectively. Scale bar, 500 μm .

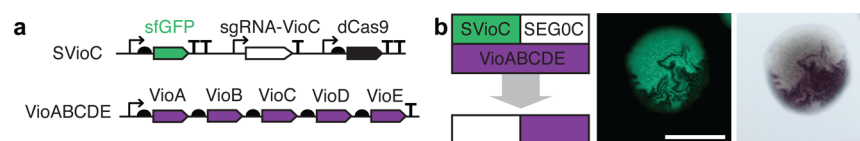


Figure 6. Domain-specific CRISPRi/Cas9 regulation of gene expression. (a) Schematic representation of spatial regulation of violacein production from VioABCDE by SVioC segregating plasmid. (b) Scale bar, 200 μm .

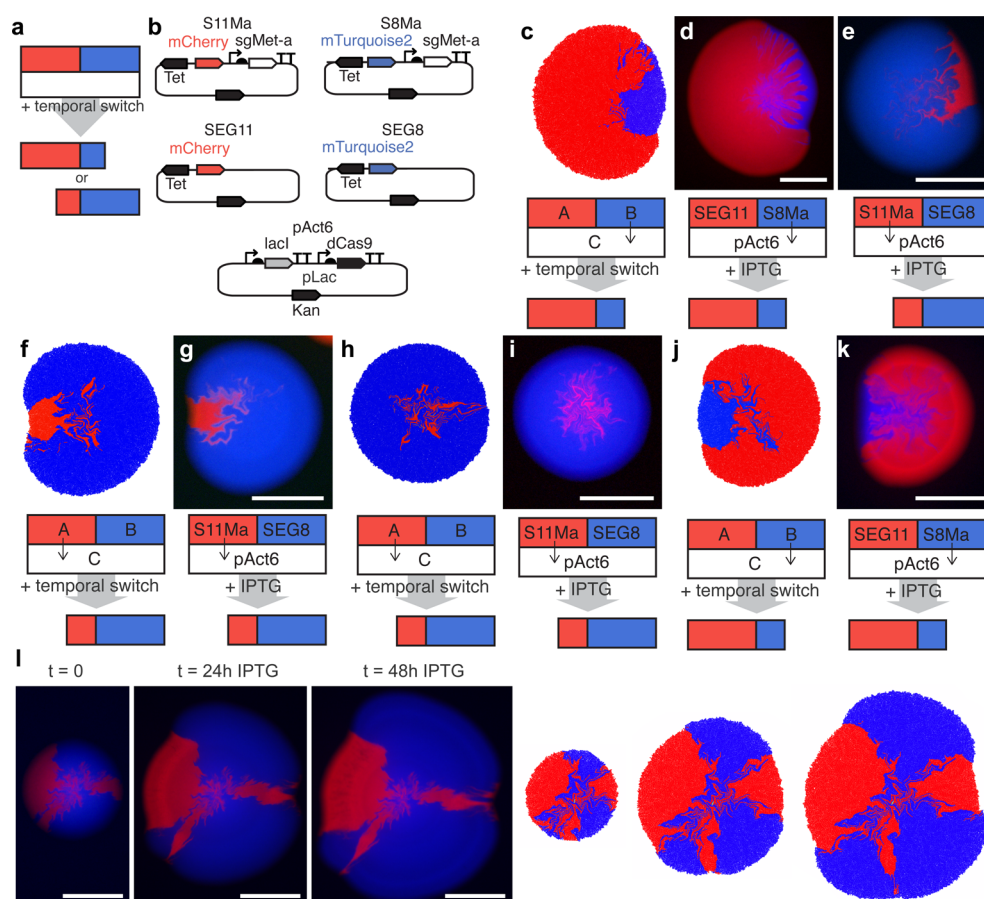


Figure 7. Morphological changes in bacterial colonies induced by symmetry-breaking and domain-specific cell growth regulation. (a) Two tier organization for morphogenetic regulation. (b) Organization of S11Ma and S8Ma plasmids containing *sgMet-a* and *pAct6* plasmid containing IPTG-regulated *dCas9* ubiquitously. (c–k) Representative images and simulations capturing the most frequent morphologies from three independent experiments. (l) Time lapse sequence of a representative colony showing the effect of initial patterns on the evolution of the morphology in both data and simulations. Scale bar, 500 μm .

Figure S9), suggesting that it could be used to induce morphological mechanism in colonies grown on solid media. Following the protocol described in previous sections, we segregated S11Ma/SEG8 or S8Ma/SEG11 in cells containing *pAct6* (Figure 7A,B), and subsequently treated them with IPTG, IPTG + methionine, or control solution. We found striking morphological features that were not obtained in control treatments (see Supporting Information Figure S10A–E). We found that the initial pattern of domains significantly influenced the final morphologies (see Supporting Information Figure S10F).

To better understand these effects, we created a CellModeller module that simulates domain-dependent cell growth in a modular organization followed from the modular design of elementary functions. We studied the whole process by simulating the same cell growth regulation step starting from different colony patterns such as bipartite (see Supporting Information Movie S4) and multisectorial (see Supporting

Information Movie S5) colonies. This facilitates the implementation and debugging of new mechanisms without affecting the rest of the components. We used CellModeller to explore *in silico* different parameters such as initial sector pattern (i.e., bipartite or multisectorial colony) and onset time for cell growth inhibition, such as triggering differential growth rates in different domains at stage 0, 1000, 5000, and 10000 cells of the simulations (see Supporting Information Movie S6 to S13). Growth rates were determined from plasmid copy numbers to range linearly between 1 (all plasmids promote growth) and 0.5 (no plasmids promote growth). We found that our models captured remarkably most of the observed morphologies in our data (Figure 7C–L). We also identified basic structural features that were over-represented in live and digital colonies (see Supporting Information Figure S10G). Together these data demonstrated that our system allows the implementation of morphogenetic instructions that give rise to reproducible features through tractable mechanisms. This system could act

as a platform to compile spatial instructions from elementary functions and features that would act as building blocks of morphogenesis.^{37,56} These could be also used to control concatenated events that can be tuned externally, similarly to the hierarchical strategies of nanofabrication inspired by biomineralization principles.⁵⁷ Ultimately, the process could be completely self-regulated by feedback signals from global emergent states. We aim to introduce cell–cell signaling to implement this global-to-local bidirectional reinforcing and feedback mechanisms with the aim of engineering self-organized morphologies and higher order dynamics.

CONCLUSIONS

Engineering morphogenesis and self-organization is increasingly gaining attention across different fields from biological sciences to robotics.^{28,37–39,58} Programming morphogenesis in biological systems is essential for constructing living functional materials,^{2,30} engineering microbial consortia with distributed functions,²⁷ and implementing distributed computation schemes.^{6,24,26} Although artificial patterning mechanisms have been already developed with remarkable results,^{21,23,25,35,36} the engineering of morphogenetic mechanisms has remained less explored.³⁸

Morphogenetic engineers face the challenge of designing functioning setups in systems with daunting complexity. The use of tractable instructions and structures could facilitate this process. We seek to apply synthetic biology tools and biophysical modeling to identify elementary functions for programming morphogenesis in simple biological systems such as bacterial colonies. Here, we created artificial mechanisms for patterning and cell differentiation that allowed the regulation of cellular states within the limits of well-defined areas of the colony. The two-tier organization to design and characterize controllers and actuators separately along with the use of modular functions abstracted into computational models in CellModeller, will be critical for gaining tractability in the future implementation of high-level designs from low-level specifications. These tools could also be combined with evolutionary computing approaches and directed evolution to explore emergent behavior and phenotype-to-genotype mappings.

Our system could also be used to benchmark models that integrate different temporal and spatial scales, a challenge for multiscale modeling. These tools could be combined with recent developments^{25,36,59} to set morphological instructions from a single cell recapitulating synthetically what developmental programs do in embryos. This will increase tractability in the process of reverse engineering emergent shapes, providing mechanistic understanding of fundamental processes underlying natural morphogenesis and development. Our platform could also be used in microbial ecology studies, for which spatial organization plays a crucial role in population dynamics.^{60–66} In turn, this could inform the implementation of new morphogenetic metaphors based on ecological functions. Starting from single cells offers the possibility of studying and engineering these phenomena from the bottom up as truly self-organized processes.

METHODS

Extended experimental procedures can be found in the Supporting Information Appendix.

DNA construction. All plasmids were constructed by Gibson assembly.⁶⁷ SEG plasmids and vectors containing each

transcriptional unit flanked with UNs were constructed from backbones pDestBAC and pJT170/2/4/6 series from Pam Silver's lab.⁵¹ pAct plasmids were constructed from backbone pSB4K5 from the Registry of Biological Parts (MIT). The VioABCDE synthetic operon was obtained from the 2009 Cambridge iGEM team (<http://2009.igem.org/Team:Cambridge/Project/Violacein>). RNAPT7, promoters and fluorescent proteins were obtained from the Registry of Parts (MIT). dCas9 was obtained from Luciano Marrafini via Addgene. All the sequences and plasmids can be found at Addgene. PCR fragments were amplified using Phusion High-Fidelity DNA Polymerase (NEB) and visualized using SYBR Safe (ThermoFisher) on a blue LED transilluminator (iorodeo.com). Purification of plasmids and PCR fragments was performed using Wizard Plus SV Minipreps DNA Purification System and Wizard SV Gel and PCR Clean-Up System (both of Promega), respectively. Primers were supplied by IDT ([idtdna.com](http://www.idtdna.com)). We used the following primers for combinatorial assemblies of UNS-flanked genes:

U1F: CATTACTCGCATCCATTCTCAGGCTG and U2R: GCTTGGATTCTGCGTTTGTTCGTC for genes flanked between UNS1 and UNS2, U2F: GCTGGGAGTTCGTAGACGGAAACAAAC and U3R: CGACCTTGATGTTTCCAGTGCGATTG for genes flanked between UNS2 and UNS3, U3F: GCACTGAAGGTCCTCAATCGCAC and U4R: GACTTTGCGTGTGTCTTACTATTGCTGG for genes flanked between UNS3 and UNS4, U4F: CTGACCTCCTGCCAGCAATAGTAAG and UXR: GGTGGAAGGGCTCGGAGTTGTGG for genes flanked between UNS4 and UNS5-UNSX.

SEG vectors for assemblies were amplified with 1XCF: GTCCTGTCTGTGACAAATTGC and U1R: GAGACGAGACGAGACAGCCTGAG; and UXF: CCAGGATACATAGATTACCACAACCTCCG and 1XCR: GAGGGCAATTGTGCACAGGGTTAAG. pAct vectors for assemblies were amplified with KAGF: TAATTACTaGTCCTTTTCCcggGAGaTcTGGGTATCTGTAAATTCTGCTAGACC and U1R: GAGACGAGACGAGACAGCCTGAG; and UXF: CCAGGATACATAGATTACCACAACCTCCG and KAGR: TACCCAGAtCTCccgGAAAAAGGACTaGTAATTATCATTGACTAGCCCATCTCAATTG.

Growth condition. All transformations were performed on *E. coli* Top10 (Invitrogen) made competent by the CCMB80 method (http://openwetware.org/wiki/TOP10_chemically_competent_cells). DH5Z1 cells were used for the segregation of three and four plasmids. Cells were grown on LB (10g of tryptone, 5g of yeast extract, and 5 g of NaCl dissolved in deionized water to a final volume of 1 L) or M9-glucose (1 × M9 salts supplemented with 2 mM MgSO₄·7H₂O, 0.1 mM CaCl₂, 0.4% glucose, and 0.2% casamino acids; 5× M9 salts contains 64 g of Na₂HPO₄·7H₂O, 15 g of KH₂PO₄, 2.5 g of NaCl, and 5 g of NH₄Cl dissolved in deionized water to a final volume of 1 L), where 1.5% w/v agar were used for solid culture. For methionine auxotrophy experiments, cells were grown in M9-glucose supplemented with L-leucine (30 mg/L). For methionine and inducer applications, we prepared a 1000× stock solution (12 g/L) of methionine, 0.1 M of IPTG, and 1 M of L-arabinose and filter sterilized them. Antibiotics were prepared as stock solutions of kanamycin (50 µg/µL), carbenicillin (100 µg/µL), tetracycline (10 µg/µL), or chloramphenicol (10 µg/µL).

Fluorescence Quantification Using Fluorometry. Fluorescence and absorbance were measured in a Clariostar plate

reader (BMG LABTECH), with GFP excited at 470–15 nm and measured at 515–20 nm, RFP excited at 570–15 nm, and measured at 620–20 nm, and absorbance measured at 600 nm.

Microscopy and Image Analysis. A Nikon Ni microscope was used for all experiments except for 4-plasmid segregation experiments. The following filter sets were used for the four fluorescent proteins used in this study: mTurquoise2 (excitation at 425–445 nm; dichroic 455 nm, emission 465–495 nm), GFP (excitation at 465–495 nm; dichroic 505 nm, emission 515–555 nm), mVenus (excitation at 490–510 nm; dichroic 515 nm, emission 520–550 nm), mCherry (excitation at 540–580 nm; dichroic 600 nm, emission 605–695 nm).

4-Plasmid segregation images were obtained using an inverted Zeiss LSM 780 multiphoton laser scanning confocal microscope. Images were taken using a 10× air objective. sfGFP was excited at 488 nm (argon ion laser), RFP was excited at 561 (HeNe laser), mTurquoise2 was excited at 458 (argon ion laser), and mVenus was excited at 488 (argon ion laser). Fluorescence was detected using a 32 anode Hybrid-GaAsP and two standard photomultiplier tubes. sfGFP emission was detected between 499 and 580 nm, RFP emission was detected between 599 and 697 nm, mTurquoise2 emission was detected between 463 and 581 nm, and mVenus emission was detected between 526 and 598 nm. The multichannel images were processed using the spectral unmixing plugin in ImageJ and then merged.⁶⁸

Image levels and brightness were applied to the whole image in Adobe Photoshop (CS6), except for those images that were used for intensity measurements.

Computational Modeling. All symmetry-breaking and domain-specific growth regulation modules were created as python scripts and run in CellModeller (<http://haselofflab.github.io/CellModeller/>). All these Python modules and MATLAB scripts used for data analysis are available at <https://github.com/timrudge/MorphoEngineering>.

■ ASSOCIATED CONTENT

■ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acssynbio.6b00149.

Extended materials and methods, as well as oligonucleotide Figures S1–S10 (PDF)
CellModeller simulations (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: ffederici@bio.puc.cl.

Author Contributions

#I.N.N., T.F.M., and I.D.D.V. contributed equally to this work

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

T.J.R., A.K., and J.H. were supported by the UK Biotechnological and Biological Sciences Research Council (BBSRC) Synthetic Biology Research Centre “OpenPlant” award (BB/L014130/1), F.F. was supported by CONICYT-PAI/Concurso Nacional de Apoyo al Retorno de Investigadores/as desde el Extranjero Folio 82130027, Fondo de Desarrollo de Areas Prioritarias (FONDAP) Center for Genome Regulation (15090007), Millennium Nucleus Center for Plant Systems and Synthetic Biology (NC130030) and Fondecyt Iniciación

11140776. A.K. was also supported by BBSRC CASE studentship in partnership with Microsoft Research. A.C. and D.E. were supported by NSF GRFP fellowship and Stanford University. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-114747 and additional support was provided by Stanford University. The authors would like to thank Rodrigo Gutierrez and his group (PUC, Chile) for support and useful comments, Joseph Torella and Pam Silver for UNSES vectors, Cambridge 2009 iGEM team for violacein operon, Registry of Biological Parts (MIT) for RNAPT7, fluorescent proteins, and pSB4K5 vector. pdCas9 was a gift from Luciano Marraffini (Addgene plasmid # 46569).

■ REFERENCES

- (1) Benenson, Y. (2012) Biomolecular computing systems: principles, progress and potential. *Nat. Rev. Genet.* 13, 455–468.
- (2) Chen, A. Y., Deng, Z., Billings, A. N., Seker, U. O., Lu, M. Y., Citorik, R. J., Zakeri, B., and Lu, T. K. (2014) Synthesis and patterning of tunable multiscale materials with engineered cells. *Nat. Mater.* 13, 515–523.
- (3) Didovyyk, A., Kanakov, O. I., Ivanchenko, M. V., Hasty, J., Huerta, R., and Tsimring, L. (2015) Distributed classifier based on genetically engineered bacterial cell cultures. *ACS Synth. Biol.* 4, 72–82.
- (4) Douglas, S. M., Bachelet, I., and Church, G. M. (2012) A logic-gated nanorobot for targeted transport of molecular payloads. *Science* 335, 831–834.
- (5) Galanie, S., Thodey, K., Trenchard, I. J., Filsinger Interrante, M., and Smolke, C. D. (2015) Complete biosynthesis of opioids in yeast. *Science* 349, 1095–1100.
- (6) Macia, J., and Sole, R. (2014) How to make a synthetic multicellular computer. *PLoS One* 9, e81248.
- (7) Nielsen, A. A., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., and Voigt, C. A. (2016) Genetic circuit design automation. *Science* 352 (6281), aac7341.
- (8) Qian, L., Winfree, E., and Bruck, J. (2011) Neural network computation with DNA strand displacement cascades. *Nature* 475, 368–372.
- (9) Siuti, P., Yazbek, J., and Lu, T. K. (2013) Synthetic circuits integrating logic and memory in living cells. *Nat. Biotechnol.* 31, 448–452.
- (10) Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R., and Benenson, Y. (2011) Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science* 333, 1307.
- (11) Ausländer, S., Ausländer, D., Müller, M., Wieland, M., and Fussenegger, M. (2012) Programmable single-cell mammalian biocomputers. *Nature* 487, 123–127.
- (12) Danino, T., Mondragón-Palomino, O., Tsimring, L., and Hasty, J. (2010) A synchronized quorum of genetic clocks. *Nature* 463, 326–330.
- (13) Del Vecchio, D. (2015) Modularity, context-dependence, and insulation in engineered biological circuits. *Trends Biotechnol.* 33, 111–119.
- (14) Kosuri, S., Goodman, D. B., Cambray, G., Mutalik, V. K., Gao, Y., Arkin, A. P., Endy, D., and Church, G. M. (2013) Composability of regulatory sequences controlling transcription and translation in *Escherichia coli*. *Proc. Natl. Acad. Sci. U. S. A.* 110, 14024–14029.
- (15) Liu, C. C., Qi, L., Lucks, J. B., Segall-Shapiro, T. H., Wang, D., Mutalik, V. K., and Arkin, A. P. (2012) An adaptor from translational to transcriptional control enables predictable assembly of complex regulation. *Nat. Methods* 9, 1088–1094.
- (16) Lou, C., Stanton, B., Chen, Y. J., Munskey, B., and Voigt, C. A. (2012) Ribozyme-based insulator parts buffer synthetic circuits from genetic context. *Nat. Biotechnol.* 30, 1137–1142.
- (17) Mutalik, V. K., Guimaraes, J. C., Cambray, G., Lam, C., Christoffersen, M. J., Mai, Q. A., Tran, A. B., Paull, M., Keasling, J. D., Arkin, A. P., and Endy, D. (2013) Precise and reliable gene expression

via standard transcription and translation initiation elements. *Nat. Methods* 10, 354–360.

(18) Qi, L., Haurwitz, R. E., Shao, W., Doudna, J. A., and Arkin, A. P. (2012) RNA processing enables predictable programming of gene expression. *Nat. Biotechnol.* 30, 1002–1006.

(19) Purnick, P. E., and Weiss, R. (2009) The second wave of synthetic biology: from modules to systems. *Nat. Rev. Mol. Cell Biol.* 10, 410–422.

(20) Balagaddé, F. K., Song, H., Ozaki, J., Collins, C. H., Barnet, M., Arnold, F. H., Quake, S. R., and You, L. (2008) A synthetic Escherichia coli predator-prey ecosystem. *Mol. Syst. Biol.* 4, 187.

(21) Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., and Weiss, R. (2005) A synthetic multicellular system for programmed pattern formation. *Nature* 434, 1130–1134.

(22) Ji, W., Shi, H., Zhang, H., Sun, R., Xi, J., Wen, D., Feng, J., Chen, Y., Qin, X., Ma, Y., Luo, W., Deng, L., Lin, H., Yu, R., and Ouyang, Q. (2013) A formalized design process for bacterial consortia that perform logic computing. *PLoS One* 8, e57482.

(23) Liu, C., Fu, X., Liu, L., Ren, X., Chau, C. K. L., Li, S., Xiang, L., Zeng, H., Chen, G., and Tang, L. H. (2011) Sequential establishment of stripe patterns in an expanding cell population. *Science Signalling* 334, 238.

(24) Regot, S., Macia, J., Conde, N., Furukawa, K., Kjellén, J., Peeters, T., Hohmann, S., de Nadal, E., Posas, F., and Solé, R. (2011) Distributed biological computation with multicellular engineered networks. *Nature* 469, 207–211.

(25) Schaeferli, Y., Munteanu, A., Gili, M., Cotterell, J., Sharpe, J., and Isalan, M. (2014) A unified design space of synthetic stripe-forming networks. *Nat. Commun.* 5, No. 4905.

(26) Tamsir, A., Tabor, J. J., and Voigt, C. A. (2011) Robust multicellular computing using genetically encoded NOR gates and chemical/wires/. *Nature* 469, 212–215.

(27) Zhou, K., Qiao, K., Edgar, S., and Stephanopoulos, G. (2015) Distributing a metabolic pathway among a microbial consortium enhances production of natural products. *Nat. Biotechnol.* 33, 377–383.

(28) Doursat, R., Sayama, H., and Michel, O. (2013) A review of morphogenetic engineering. *Nat. Comput.* 12, 517–535.

(29) Doursat, R., Sayama, H., and Michel, O. (2012) *Morphogenetic Engineering: Toward Programmable Complex Systems*, Springer.

(30) Chen, A. Y., Zhong, C., and Lu, T. K. (2015) Engineering living functional materials. *ACS Synth. Biol.* 4, 8–11.

(31) Ball, P., and Borley, N. R. (1999) *The self-made tapestry: pattern formation in nature*, Oxford University Press, Oxford.

(32) Goodwin, B. C., Kauffman, S., and Murray, J. D. (1993) Is morphogenesis an intrinsically robust process? *J. Theor. Biol.* 163, 135–144.

(33) Bashor, C. J., Horwitz, A. A., Peisajovich, S. G., and Lim, W. A. (2010) Rewiring cells: synthetic biology as a tool to interrogate the organizational principles of living systems. *Annu. Rev. Biophys.* 39, 515.

(34) Holland, J. H. (2012) *Signals and boundaries: Building blocks for complex adaptive systems*, Mit Press.

(35) Grant, P. K., Dalchau, N., Brown, J. R., Federici, F., Rudge, T. J., Yordanov, B., Patange, O., Phillips, A., and Haseloff, J. (2016) Orthogonal intercellular signaling for programmed spatial behavior. *Mol. Syst. Biol.* 12, 849.

(36) Payne, S., Li, B., Cao, Y., Schaeffer, D., Ryser, M. D., and You, L. (2013) Temporal control of self-organized pattern formation without morphogen gradients in bacteria. *Mol. Syst. Biol.* 9, 697.

(37) Cachat, E., Liu, W., Hohenstein, P., and Davies, J. A. (2014) A library of mammalian effector modules for synthetic morphology. *J. Biol. Eng.* 8, 1.

(38) Pascualie, J., Potier, M., Kowaliw, T., Giavitto, J.-L., Michel, O., Spicher, A., and Doursat, R. (2016) Developmental design of synthetic bacterial architectures by morphogenetic engineering. *ACS Synth. Biol.* 5, 842–61.

(39) Davies, J. A. (2008) Synthetic morphology: prospects for engineered, self-constructing anatomies. *J. Anat.* 212, 707–719.

(40) Foty, R. A., and Steinberg, M. S. (2005) The differential adhesion hypothesis: a direct evaluation. *Dev. Biol.* 278, 255–263.

(41) Bowman, J. L., Smyth, D. R., and Meyerowitz, E. M. (2012) The ABC model of flower development: then and now. *Development* 139, 4095–4098.

(42) Rudge, T. J., Steiner, P. J., Phillips, A., and Haseloff, J. (2012) Computational modeling of synthetic microbial biofilms. *ACS Synth. Biol.* 1, 345–352.

(43) Rudge, T. J., Federici, F., Steiner, P. J., Kan, A., and Haseloff, J. (2013) Cell Polarity-Driven Instability Generates Self-Organized, Fractal Patterning of Cell Layers. *ACS Synth. Biol.* 2, 705–714.

(44) Anderson, P. W. (1972) More is different. *Science* 177, 393–396.

(45) Gierer, A., and Meinhardt, H. (1972) A theory of biological pattern formation. *Biological Cybernetics* 12, 30–39.

(46) Munro, E., and Bowerman, B. (2009) Cellular symmetry breaking during Caenorhabditis elegans development. *Cold Spring Harbor Perspect. Biol.* 1, a003400.

(47) Tompkins, N., Li, N., Girabawe, C., Heymann, M., Ermentrout, G. B., Epstein, I. R., and Fraden, S. (2014) Testing Turing's theory of morphogenesis in chemical cells. *Proc. Natl. Acad. Sci. U. S. A.* 111, 4397–4402.

(48) Turing, A. M. (1952) The chemical basis of morphogenesis. *Philos. Trans. R. Soc., B* 237, 37–72.

(49) van der Gucht, J., and Sykes, C. (2009) Physical model of cellular symmetry breaking. *Cold Spring Harbor Perspect. Biol.* 1, a001909.

(50) Wennekamp, S., Mesecke, S., Nédélec, F., and Hiragi, T. (2013) A self-organization framework for symmetry breaking in the mammalian embryo. *Nat. Rev. Mol. Cell Biol.* 14, 452–459.

(51) Torella, J. P., Boehm, C. R., Lienert, F., Chen, J. H., Way, J. C., and Silver, P. A. (2014) Rapid construction of insulated genetic circuits via synthetic sequence-guided isothermal assembly. *Nucleic Acids Res.* 42, 681–689.

(52) Chen, Y. J., Liu, P., Nielsen, A. A., Brophy, J. A., Clancy, K., Peterson, T., and Voigt, C. A. (2013) Characterization of 582 natural and synthetic terminators and quantification of their design constraints. *Nat. Methods* 10, 659–664.

(53) Dahmann, C., Oates, A. C., and Brand, M. (2011) Boundary formation and maintenance in tissue development. *Nat. Rev. Genet.* 12, 43–55.

(54) August, P. R., Grossman, T. H., Minor, C., Draper, M. P., MacNeil, I. A., Pemberton, J. M., Call, K. M., Holt, D., and Osburne, M. S. (2000) Sequence analysis and functional characterization of the violacein biosynthetic pathway from Chromobacterium violaceum. *J. Mol. Microbiol. Biotechnol.* 2, 513–519.

(55) Segall-Shapiro, T. H., Meyer, A. J., Ellington, A. D., Sontag, E. D., and Voigt, C. A. (2014) A 'resource allocator' for transcription based on a highly fragmented T7 RNA polymerase. *Mol. Syst. Biol.* 10, 742.

(56) Newman, S. A., and Bhat, R. (2009) Dynamical patterning modules: a "pattern language" for development and evolution of multicellular form. *Int. J. Dev. Biol.* 53, 693–705.

(57) Noorduyn, W. L., Grinthal, A., Mahadevan, L., and Aizenberg, J. (2013) Rationally designed complex, hierarchical microarchitectures. *Science* 340, 832–837.

(58) Werfel, J., Petersen, K., and Nagpal, R. (2014) Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 754–758.

(59) Isalan, M., Lemerle, C., and Serrano, L. (2005) Engineering gene networks to emulate Drosophila embryonic pattern formation. *PLoS Biol.* 3, e64.

(60) Korolev, K. S., Müller, M. J., Karahan, N., Murray, A. W., Hallatschek, O., and Nelson, D. R. (2012) Selective sweeps in growing microbial colonies. *Phys. Biol.* 9, 026008.

(61) Blanchard, A. E., and Lu, T. (2015) Bacterial social interactions drive the emergence of differential spatial colony structures. *BMC Syst. Biol.* 9, 59.

- (62) Brenner, K., and Arnold, F. H. (2011) Self-organization, layered structure, and aggregation enhance persistence of a synthetic biofilm consortium. *PLoS One* 6, e16791.
- (63) Hallatschek, O., Hersen, P., Ramanathan, S., and Nelson, D. R. (2007) Genetic drift at expanding frontiers promotes gene segregation. *Proc. Natl. Acad. Sci. U. S. A.* 104, 19926–19930.
- (64) Hauert, C., and Doebeli, M. (2004) Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nat. Mater.* 428, 643–646.
- (65) Kovács, T. (2014) Impact of spatial distribution on the development of mutualism in microbes. *Front Microbiol.* 5, 649.
- (66) Hol, F. J., Galajda, P., Woolthuis, R. G., Dekker, C., and Keymer, J. E. (2015) The idiosyncrasy of spatial structure in bacterial competition. *BMC Res. Notes* 8, 245.
- (67) Gibson, D. G., Young, L., Chuang, R. Y., Venter, J. C., Hutchison, C. A., and Smith, H. O. (2009) Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* 6, 343–345.
- (68) Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012) NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* 9, 671–675.

ANEXO B: RUTINAS DE MATLAB

A continuación, se presentan los códigos de las rutinas MATLAB indicadas en la sección 3.6 (materiales y métodos). Con ellos se efectuaron los análisis de las imágenes de microscopía obtenidas en los ensayos de cambios morfológicos en las colonias bacterianas.

convertSingleImagesToRGB.m

```
% function convertImagesToRGB(dirname);
function convertSingleImagesToRGB()           #runs the function on working space folder

% Go through images and convert it to 8 bit
% Identify if the file is Red fluorescent channel, Blue fluorescent channel or Brightfield
% save as a RGB .tif file with an appropriate tag: Red (R), Blue(B), Brightfield (F).

imfiles = dir('*.tif');
nfiles = size(imfiles);
for i=1:nfiles;
    namefile = imfiles(i).name;

    % Read images, scale to max, and convert to 8 bit
    file = imread(namefile);
    file = double(file)/double(max(file(:)));
    file = file*255;

    im = uint8(file);

    % identify channel
    ChannelValue = zeros(3,1);    %variable initialization

    for j=1:3
        ChannelValue(j) = sum(sum(im(:, :, j))); %sum all the values on the current channel
    end

    maxV=max(ChannelValue);

    channel=1; %channel auxiliar index
    for a=1:3
        if maxV == ChannelValue(a)
            channel=a;
        end
    end

    % write new file
    if channel == 1
        ofname = [namefile, '_R.tif'];
        imwrite(im, ofname);

    elseif channel == 3
        ofname = [namefile, '_B.tif'];
        imwrite(im, ofname);

    else
        ofname = [namefile, '_F.tif'];
```



```

        imwrite(im, ofname);
    end
end;

```

mergeImagesToRGB.m

```

% function convertImagesToRGB(dirname);
function mergeImagesToRGB() #runs the function on working space folder

% Go through images ordered red, blue, brightfield
% Load red and blue images, merge them, and save as a RGB .tif file

imfiles = dir('*.tif');
nfiles = size(imfiles);

mkdir('merged') %create a folder to save the merged files

for i=1:4:nfiles;

    for j=0:3
        if isempty(strfind(imfiles(i+j).name,'R'))== 0
            rfile = imfiles(i+j).name;

            elseif isempty(strfind(imfiles(i+j).name,'B'))== 0
                bfile = imfiles(i+j).name;

        end
    end

    % Read images
    red = imread(rfile);
    blue = imread(bfile);

    % .tif supressed from the name
    rfile=strtok(rfile,'.');
    bfile=strtok(bfile,'.');

    % file write
    im = uint8(red+blue);
    ofname = ['M_',rfile, '_', bfile, '.tif'];
    imwrite(im, ofname);

    % move the file to the new directory 'merged'
    movefile([pwd,'\ ',ofname],[pwd,'\merged\ ',ofname])
end;

```

ColonySelection.m

```

function ColonySelection
% Go through images in a folder to let the user select desired colonies on each image.
% Save the selected point and the name of the image in an array.

imfiles = dir('*.tif');
nfiles = size(imfiles);

Data = cell(nfiles,3); % {name x y}

for i=1:nfiles;

    nfile = imfiles(i).name;

```

```

    Data(i,1)={nfile};    % Save name

    image=imread(nfile);
    imshow(image);
    title(nfile);
    [x y]=ginput(1);      % Press left-click anywhere inside the selected colony
    close

    Data(i,2)={x};
    Data(i,3)={y};
end

save([pwd,'\ColPos'],'Data')    % save file with colony positions

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

AllParameters.m

imfiles = dir('*.tif');
nfiles = size(imfiles);
load('ColPos.mat')
%%%
% these lines are only necessary the first time the code is executed

Params = cell(nfiles(1,1),5);           %{name gsize gsigma Rthresh Bthresh}
save([pwd,'\Parameters'],'Params')      %create the file
Values = cell(nfiles(1,1),3);           %{name circ ecc}
save([pwd,'\MorphVal'],'Values')

%%%

load('Parameters.mat')
load('MorphVal.mat')

for i=1:nfiles;
    nfile = imfiles(i).name;

    x = Data{i,2};
    y = Data{i,3};

    if isempty(Params{i,5})== 1

        gsigma0=4;
        gsize0=6*gsigma0+1;
        Rthresh0=0;
        Bthresh0=0;

        [gsize,gsigma,Rthresh, Bthresh, circ, ecc] = getParameters(nfile,x,y,gsize0, gsigma0,
        Rthresh0,Bthresh0);

        Params(i,1)={nfile};
        Params(i,2)={gsize};
        Params(i,3)={gsigma};
        Params(i,4)={Rthresh};
        Params(i,5)={Bthresh};

        save([pwd,'\Parameters'],'Params')

        Values(i,1)={nfile};
        Values(i,2)={circ};
        Values(i,3)={ecc};
    end
end

```

```

        save([pwd, '\MorphVal'], 'Values')

        Q = 'N'; % assing a negative connotation to Q
    else
        Q=input(sprintf('Do you want to check the parameters of %s , (file %d) [Y/N] ?', nfile, i),
's');
    end

    if Q == 'Y'
        gsize0=Params{i,2};
        gsigma0=Params{i,3};
        Rthresh0=Params{i,4};
        Bthresh0=Params{i,5};

        [gsize,gsigma,Rthresh, Bthresh, circ, ecc] = getParameters(nfile,x,y, gsize0, gsigma0);

        Params(i,1)={nfile};
        Params(i,2)={gsize};
        Params(i,3)={gsigma};
        Params(i,4)={Rthresh};
        Params(i,5)={Bthresh};

        save([pwd, '\Parameters'], 'Params')

        Values(i,1)={nfile};
        Values(i,2)={circ};
        Values(i,3)={ecc};

        save([pwd, '\MorphVal'], 'Values')

    else

    end

end

save([pwd, '\Parameters'], 'Params') % save file with parameters for every image
save([pwd, '\MorphVal'], 'Values') % save computed values

```

getParameters

```

function [gsize,gsigma,Rthresh, Bthresh, circ, ecc] = getParameters(fname, x, y, gsize0, gsigma0,
Rthresh0, Bthresh0)

% Shape analysis for morphological engineering
% fname = image file
% x and y are a pixel inside selected colony to demark its position
%
% Returns:
% thresh = threshold for normalised (0-1) image, e.g. 0.2
% gsize = size of Gaussian filter (pixels), e.g. [9,9]
% gsigma = standard deviation of Gaussian filter (pixels), e.g. 1.5
% circ = circularity
% ecc = eccentricity
% pRed = red area / total area
% pGreen = green area / total area
% pBlue = blue area / total area

```

```

%%
% WARNING
% as no saved thresh parameters exist, counter2 start on value=2.
%%

im0=imread(fname);
fname
fh = figure(1);
figure(fh); subplot(2,2,1); imshow(im0); axis image; title('Original image');

reply='N'; %remains 'N' meanwhile the user consider parameters are wrong
imd=double(im0);

%%%%%%%%
% Smooth image %
%%%%%%%%

counter=1;

while reply ~= 'Y'

    replyS='N';
    while replyS ~= 'Y'
        if counter == 1
            gsize=gsize0; % this assignation is important for figure
purposes
            gsigma=gsigma0;
            h = fspecial('gaussian', gsize, gsigma);
            im = imfilter(imd, h);
            % uncomment these lines to check the filter effect on original image
            im0f = imfilter(im0,h);
            % figure(2); subplot(2,4,[3:4 7:8]); imshow(im0f); axis image; title('filtered
image');
        else
            % figure(fh); subplot(2,2,4); imagesc(bimf); axis image; colorbar; title(['Last
thresholded image - ', ['Red thresh: ', num2str(Rthresh)] ,[' , Blue thresh: ', num2str(Bthresh)]]);

            % uncomment this line to define manually the filter size
            % gsiz=input(sprintf('Last value = %d \n enter size of Gaussian filter (pixels, value
from 1 to 100): ', gsize(1,1)));

            gsigma=input(sprintf('Last value = %d \n enter standard deviation of Gaussian filter
(pixels, value from 1 to 10): ', gsigma));
            gsiz=gsigma*6+1; # default value assignation to filter size, accord sigma
            gsize=[gsiz,gsiz];

            h = fspecial('gaussian', gsize, gsigma);
            im = imfilter(imd, h);

            im0f = imfilter(im0,h);

        end
        counter = 2; %only the first time is runned with default parameters

        % Display image that will be thresholded
        figure(fh); subplot(2,2,2); imshow(im0f); axis image; title(['Smooth image for
thresholding - ', [' gsize: ', num2str(gsize(1,1))] ,[' , gsigma: ', num2str(gsigma)]]);

        replyS = input('Smooth parameters are OK? [Y/N]: ', 's');
    end

    %%%%%%%%%
    %%%%%%%%% Threshold %%%%%%%%%
    %%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
counter2=1; % if counter2=1, saved values are used to perform the threshold

replyT='N';
Rthresh=Rthresh0;
Bthresh=Bthresh0;
while replyT ~= 'Y'

    % Assign threshold
    if counter2 ==1 %use saved threshold values (the first loop)

        Rthresh=Rthresh0;
        Bthresh=Bthresh0;

        thresh = thresh0; % default threshold value
        bimf = (bim>thresh);

    else % to assign new threshold values

        SizeIm=size(im);

        %transects
        inty=floor(y);
        Rline=im(inty,:,1);
        Bline=im(inty,:,3);

        % figure to select threshold values
        figure(2); subplot(2,4,[1:2 5:6]); plot(Rline,'ro'); title('Red transect intensity
profile');
        figure(2); subplot(2,4,[3:4 7:8]); plot(Bline,'bo'); title('Blue transect intensity
profile');

        repTresh='N';

        while repTresh ~= 'Y'

            Rthresh=input(sprintf('Last Red value = %d \n threshold for Red: ',Rthresh));
            Bthresh=input(sprintf('Last Blue value = %d \n threshold for Blue: ',Bthresh));

            RthreshLine=ones(1,SizeIm(2))*Rthresh;
            BthreshLine=ones(1,SizeIm(2))*Bthresh;

            % plot threshold line

            figure(2); subplot(2,4,[1:2 5:6]); plot(Rline,'ro'); title('Red transect
intensity profile');
            hold on
            figure(2); subplot(2,4,[3:4 7:8]); plot(Bline,'bo'); title('Blue transect
intensity profile');
            hold on
            figure(2);subplot(2,4,[1:2 5:6]); plot(RthreshLine,'k'); hold off;
            subplot(2,4,[3:4 7:8]); plot(BthreshLine,'k'); hold off

            repTresh = input('Threshold lines are OK? [Y/N]: ', 's');
        end

        % Apply threshold

        Tim=zeros(SizeIm(1),SizeIm(2),3); % Empty matrix to save thresholded image

        for i=1:SizeIm(1)

```

```

        for j= 1:SizeIm(2)
            if im(i,j,1)> Rthresh
                Tim(i,j,1) = im(i,j,1);
            end
            if im(i,j,3)> Bthresh
                Tim(i,j,3) = im(i,j,3);
            end
        end
    end
end

end
counter2 = 2; % only the first time is executed with default (or saved) parameters

% Thresholded image for overall morphology (1 layer)
[Nim, chidx] = max(Tim, [], 3); % Take max of (r,g,b) channels to get overall shape,
chidx is the channel num that is max
Nim = (Nim-min(Nim(:)))/max(Nim(:)); % normalise

% Display thresholded image (smooth values)
figure(fh); subplot(2,2,2); imagesc(Nim); axis image; colorbar; title(['Normalised
thresholded image -',' gsize: ',num2str(gsize(1,1))] ,[' ', gsize: ',num2str(gsize)]]);

% Display thresholded binary image (binary values)
bimf=(Nim>0);
figure(fh); subplot(2,2,3); imagesc(bimf); axis image; colorbar; title(['Normalised
thresholded binary image -','Red thresh: ',num2str(Rthresh)] ,[' ', Blue thresh:
',num2str(Bthresh)]]);

% plot blue pixels and red pixels separated
RTim=(Tim(:,:,1)>0); %Binary image of red pixels
BTim=(Tim(:,:,3)>0); %Binary image of blue pixels

%Eliminate pixels not selected
for i=1:SizeIm(1)
    for j= 1:SizeIm(2)
        if chidx(i,j) ~= 1
            RTim(i,j)=0;
        end
        if chidx(i,j) ~= 3
            BTim(i,j)=0;
        end
    end
end

figure(3); subplot(2,1,1); imagesc(RTim); axis image; colorbar; title('Red binary
image');
figure(3); subplot(2,1,2); imagesc(BTim); axis image; colorbar; title('Blue binary
image');
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Isolate colony and metrics %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% isolate desired object on image == colony
[bim, idx]=bwselect(bimf,x,y,8);
% figure(1)
% imshow(bim)
% title(fname)

% Display final single selected connected component (colony) image
figure(fh); subplot(2,2,4); imagesc(bim); axis image; colorbar; title(['single selected
connected component - ', ['size: ', num2str(gsize(1,1))] , ' , Rthresh ', num2str(Rthresh)], [' ,
Bthresh: ', num2str(Bthresh)]);

% Get some measures of the shape
rp = regionprops(bim, 'all');

% There might be multiple colonies - we should avoid this, split images
nc = length(rp);
circ = []; ecc = [];
for i=1:nc;
    c = 4*pi*rp(i).Area/(rp(i).Perimeter^2);
    circ = [circ, c];
    ecc = [ecc, rp(i).Eccentricity];
end;

disp(sprintf('circularity = %d', circ))
disp(sprintf('eccentricity = %d', ecc))

%user approve
replyT = input('Threshold parameters are OK? [Y/N]: ', 's');
end

% Uncomment this lines to perform area computations

% % Area ratios for each colour
% bch = chidx.*bim; % Index of maximum channel, masked by threshold for overall morphology
% Ared = sum(bch(:)==1);
% Agreen = sum(bch(:)==2);
% Ablue = sum(bch(:)==3);
% Atot = Ared + Ablue + Agreen;
%
% pRed = Ared/Atot;
% pBlue = Ablue/Atot;
%
% fprintf('pRed = %d \n pBlue = %d \n',pRed,pBlue)

% final question
reply = input('All parameters are OK? [Y/N]: ', 's'); % if you are conform with parameters
press Y
end

```

IndividualAnalysis.mat

```

% perform smooth and isolate and colony on a desired image
% helps to debug and explore the analysis

```

```

fname='M_0139_0140.tif'; % name of the desired image
imageA=imread(fname);

```

```

imshow(imageA);
[x y]=ginput(1);
close

im=double(imageA);
fname

% Smooth image
h = fspecial('gaussian', [21,21], 3);
im = imfilter(im, h);

% Binary image for overall morphology
[bim, chidx] = max(im, [], 3); % Take max of (r,g,b) channels to get overall shape, chidx is the
channel num that is max
bim = (bim-min(bim(:)))/max(bim(:)); % normalise
bim = (bim>0.3); % threshold at 10%

% isolate desired object on image == colony
[bim, idx]=bwselect(bim,x,y,8);
figure(1)
imshow(bim)
title(fname)

MorphStatics.m

% script to analyze morphological data

% Four groups were analyzed: (G1) control with IPTG added at time=0h but enriched with %
methionine, (G2) IPTG added at time= 0h, (G3) added at time=8h, and (G4) without %induction.

% Morphological used measures are:
% Circularity: circle=1, noncircle=0.
% Eccentricity: circle=0, elipse=1.

% Steps:
% - Load Data, and add it in order to AllValues matrix.
% (every saved data have a boolean vector to say if the colony is used in the analisys or not)
% - Mean and standar deviation is calculated for each measure of every group
% - bar plots are made
% - Anova is performed for circularity and eccentricity

function [AllValues MorphMean MorphStd]=MorphStatic
% Returns:
% AllValues: Matrix with circularity and eccentricity data for the 4 groups.
%           [circ(G1) circ(G2) circ(G3) circ(G4) ecc(G1) ecc(G2) ecc(G3) ecc(G4)]
% MorphMean: Vector with the mean value of circularity and eccentricity for
%           each group. (same order as AllValues)
% MorphStd:  vector with standard deviation of circularity and eccentricity for
%           each group. (same order as AllValues)

AllValues = zeros(1,8);          % initialize matrix

load('MorphVal0h+met.mat') % (G1)
c=1;
for i=1:length(Values)
    if Values{i,4}>0
        AllValues(c,1)=Values{i,2}*Values{i,4};
        AllValues(c,5)=Values{i,3}*Values{i,4};
        c=c+1;
    end
end
end

```



```

load('MorphVal0h.mat')      %(G2)
c=1;
for i=1:length(Values)
    if Values{i,4}>0
        AllValues(c,2)=Values{i,2}*Values{i,4};
        AllValues(c,6)=Values{i,3}*Values{i,4};
        c=c+1;
    end
end

load('MorphVal8h.mat')      %(G3)
c=1;
for i=1:length(Values)
    if Values{i,4}>0
        AllValues(c,3)=Values{i,2}*Values{i,4};
        AllValues(c,7)=Values{i,3}*Values{i,4};
        c=c+1;
    end
end

load('MorphValnever.mat')    %(G4)
c=1;
for i=1:length(Values)
    if Values{i,4}>0
        AllValues(c,4)=Values{i,2}*Values{i,4};
        AllValues(c,8)=Values{i,3}*Values{i,4};
        c=c+1;
    end
end

AllValues(AllValues==0)=NaN;      %make NaN the empty spots.
MorphMean=nanmean(AllValues,1)
MorphStd=nanstd(AllValues,1);

X=(1:1:4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot results %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%Circularity
%

figure(1)

%for i=1:4
%plot(X(i),AllValues(:,i),'ro')    %all points
%bar(X,MorphMean(1:4))             %for Y linear-scale plot

bar(X,MorphMean(1:4),'baseValue',0.1)
set(gca,'XTick',1:1:4)
set(gca,'XTickLabel',{'IPTG 0h + Met','IPTG 0h','IPTG 8h','w/o IPTG'})
set(gca,'YScale','log')
ylim([0.85 0.91])
title('Circularity')
hold on
%end
errorbar(X,MorphMean(1:4),zeros(1,4),MorphStd(1:4),'.b')
ylabel('circularity')
xlabel('Group')
hold off

```

```

% Eccentricity

figure(2)
%for i=1:4
%plot(X(i),AllValues(:,i+4),'ro') %all points
bar(X,MorphMean(5:8))
set(gca,'XTick',1:1:4)
set(gca,'XTickLabel',{'IPTG 0h + Met','IPTG 0h','IPTG 8h','w/o IPTG'})
title('Eccentricity')
hold on
%end
errorbar(X,MorphMean(5:8),zeros(1,4),MorphStd(5:8),'.b')
ylabel('eccentricity')
xlabel('Group')
hold off

%ANOVA of Circularity

[p,tbl,stats] = anova1(AllValues(:,1:4));
set(gca,'XTickLabel',{'IPTG 0h + Met','IPTG 0h','IPTG 8h','w/o IPTG'})

%Multiple Comparision Test, to know what mean are diferents
stats.gnames(:,1)={'IPTG 0h + Met','IPTG 0h','IPTG 8h','w/o IPTG'};
figure()
[c,m,h,nms] =multcompare(stats);
[nms num2cell(m)]

%ANOVA of Eccentricity

[p,tbl,stats] = anova1(AllValues(:,5:8));
set(gca,'XTickLabel',{'IPTG 0h + Met','IPTG 0h','IPTG 8h','w/o IPTG'})

%Multiple Comparision Test, to know what mean are diferents

stats.gnames(:,1)={'IPTG 0h + Met','IPTG 0h','IPTG 8h','w/o IPTG'};
figure()
[c,m,h,nms] =multcompare(stats);
[nms num2cell(m)]

% %%%
% %MANOVA to use Circularity and eccentricity
% n=size(AllValues,1);
% m=size(AllValues,2);
% VectorValues=zeros(4*n,2);
% group=zeros(m*n/2,1);
% for j=1:m
%     if j<5
%         VectorValues(((j-1)*n+1):(j*n),1)=AllValues(:,j);
%     else
%         VectorValues(((j-5)*n+1):((j-4)*n),2)=AllValues(:,j);
%     end
% end
% for i=1:4
%     group(((i-1)*n+1):(n*i))=ones(n,1)*i;
% end
% [d,pMan,statsMan] = manova1(VectorValues,group);
% d
% pMan

```

ANEXO C: RUTINAS USADAS EN CELLMODELLER

Los siguientes códigos ejemplifican la implementación y uso del módulo desarrollado para representar la ruptura de simetría. Estos fueron ejecutados directamente en CellModeller.

Código 1: Ruptura de simetría a partir de colonia fundadora con una copia de dos plásmidos distintos en su interior

```
import random
from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
from CellModeller.GUI import Renderers
from CellModeller.Integration.CLCrankNicIntegrator import CLCrankNicIntegrator #add
import numpy
import math

#Specify parameter for solving diffusion dynamics
grid_dim = (64, 8, 12) # dimension of diffusion space, unit = number of grid
grid_size = (4, 4, 4) # grid size
grid_orig = (-128, -14, -8) # where to place the diffusion space onto simulation space

def setup(sim):
    # Set biophysics, signalling, and regulation models
    biophys = CLBacterium(sim, jitter_z=False, max_cells=30000,max_sqs=192**2)

    regul = ModuleRegulator(sim, sim.moduleName)# use this file for reg too
    # Only biophys and regulation
    sim.init(biophys, regul, None, None)

    sim.addCell(cellType=0, pos=(0,0,0))

    # Add some objects to draw the models
    therenderer = Renderers.GLBacteriumRenderer(sim)
    sim.addRenderer(therenderer)
    sim.pickleSteps = 10

def numSpecies():
    return(0)

def numSignals():
    return(0)

def init(cell):
    cell.targetVol = 3.5 + random.uniform(0.0,0.5)
    cell.growthRate = 1.0
    cell.n_a = 1
    cell.n_b = 1

def update(cells):
    for (id, cell) in cells.iteritems():
        if len(cells)>30000:
            gr1 = 1.0
            gr2 = 0.5
            cell.growthRate = (gr2-gr1)*cell.n_a*0.5 + gr1
```

```

cell.color = [0.1, cell.n_a/2.0, cell.n_b/2.0]
if cell.volume > cell.targetVol:
    cell.divideFlag = True

def divide(parent, d1, d2):
    d1.targetVol = 3.5 + random.uniform(0.0,0.5)
    d2.targetVol = 3.5 + random.uniform(0.0,0.5)
    plasmids = [0]*parent.n_a*2 + [1]*parent.n_b*2
    random.shuffle(plasmids)
    d1.n_a = 0
    d1.n_b = 0
    d2.n_a = 0
    d2.n_b = 0
    for p in plasmids[:2]:
        if p == 0: d1.n_a +=1
        else: d1.n_b +=1
    for p in plasmids[2:4]:
        if p == 0: d2.n_a +=1
        else: d2.n_b +=1
    assert parent.n_a + parent.n_b == 2
    assert d1.n_a + d1.n_b == 2
    assert d2.n_a + d2.n_b == 2
    assert parent.n_a*2 == d1.n_a+d2.n_a
    assert parent.n_b*2 == d1.n_b+d2.n_b
    assert parent.n_a > 0 or (d1.n_a == 0 and d2.n_a == 0)
    assert parent.n_b > 0 or (d1.n_b == 0 and d2.n_b == 0)

```

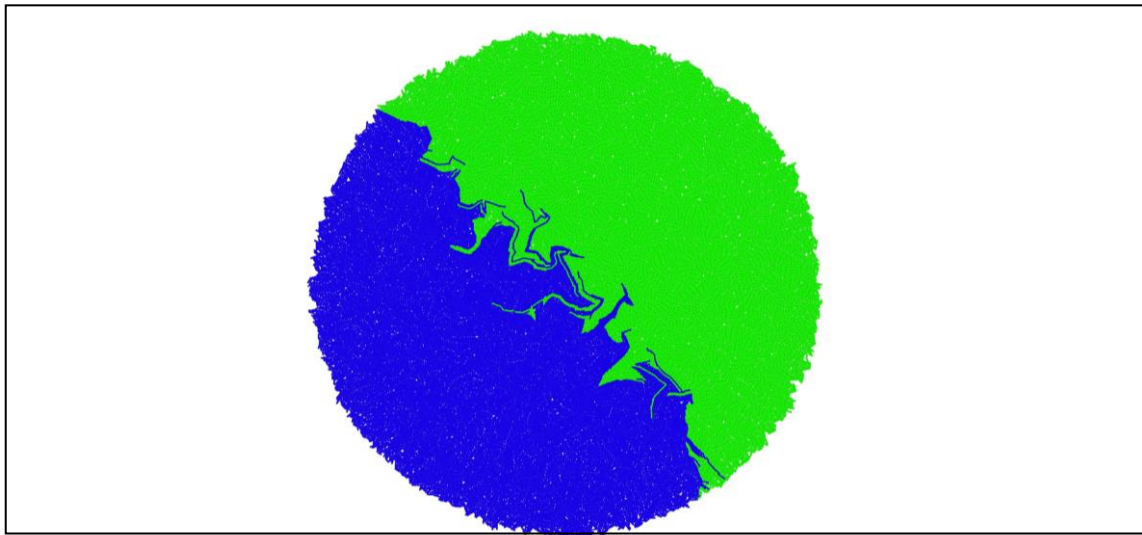


Figura A- 1: Ruptura de simetría a partir de célula fundadora con dos plásmidos distintos en su interior. La imagen muestra el estado final de una simulación efectuada con el código 1.

Código 2: Ruptura de simetría a partir de colonia fundadora con una copia de tres plásmidos distintos en su interior

```

import random
from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
from CellModeller.GUI import Renderers
from CellModeller.Integration.CLCrankNicIntegrator import CLCrankNicIntegrator #add
import numpy
import math

def setup(sim):
    # Set biophysics, signalling, and regulation models
    biophys = CLBacterium(sim, jitter_z=False, max_cells=50000)

    regul = ModuleRegulator(sim, sim.moduleName)# use this file for reg too
    # Only biophys and regulation
    sim.init(biophys, regul, None, None)

    sim.addCell(cellType=0, pos=(0,0,0))

    # Add some objects to draw the models
    therenderer = Renderers.GLBacteriumRenderer(sim)
    sim.addRenderer(therenderer)
    sim.pickleSteps = 10

def init(cell):
    cell.targetVol = 3.5 + random.uniform(0.0,0.5)
    cell.growthRate = 1.0
    cell.n_a = 1
    cell.n_b = 1

def init(cell):

    cell.targetVol = 3.5 + random.uniform(0.0,0.5)

    cell.growthRate = 1.0

    cell.n = [1,1,1] # 1 copy of 3 different plasmids

def update(cells):

    for (id, cell) in cells.iteritems():

        r = 0.3+cell.n[0]/3.0

        g = 0.3+cell.n[1]/3.0

        b = 0.3+cell.n[2]/3.0

        cell.color = [r,g,b]

        if cell.volume > cell.targetVol:

            cell.divideFlag = True

def divide(parent, d1, d2):

```

```

d1.targetVol = 3.5 + random.uniform(0.0,0.5)
d2.targetVol = 3.5 + random.uniform(0.0,0.5)

plasmids = []
for i in range(3):
    plasmids += [i]*parent.n[i]*2
    d1.n[i] = 0
    d2.n[i] = 0
random.shuffle(plasmids)
for p in plasmids[:3]:
    d1.n[p] += 1
for p in plasmids[3:6]:
    d2.n[p] += 1

```

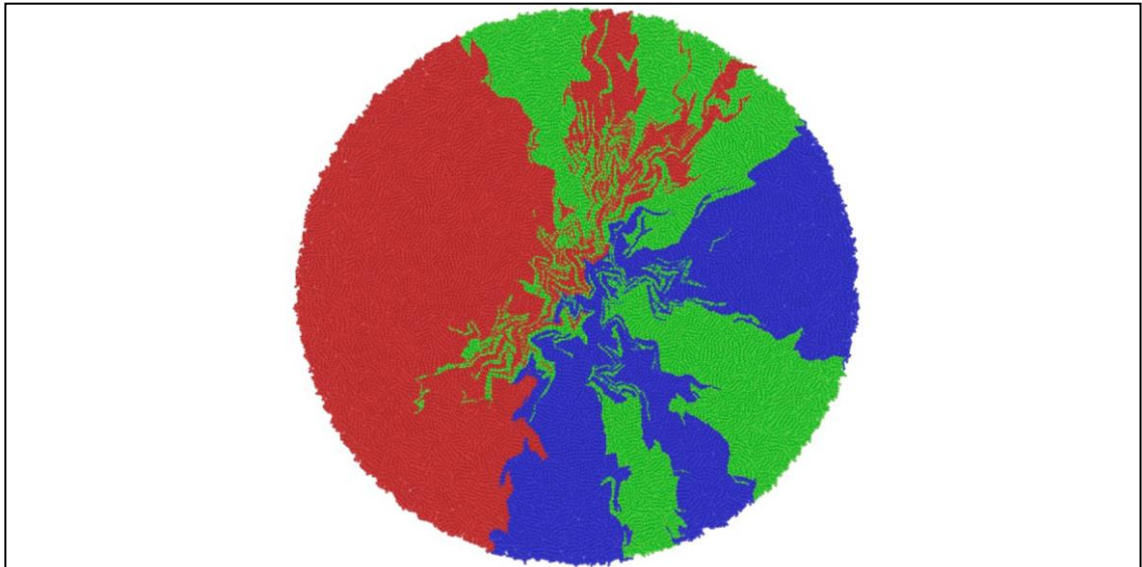


Figura A- 2: Ruptura de simetría a partir de célula fundadora con tres plásmidos distintos en su interior. La imagen muestra el estado final de una simulación efectuada con el código 2.

Código 3: Ruptura de simetría a partir de colonia fundadora con una copia de cuatro plásmidos distintos en su interior

```

import random
from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
from CellModeller.GUI import Renderers
from CellModeller.Integration.CLCrankNicIntegrator import CLCrankNicIntegrator #add
import numpy
import math

def setup(sim):
    # Set biophysics, signalling, and regulation models
    biophys = CLBacterium(sim, jitter_z=False, max_cells=50000)

    regul = ModuleRegulator(sim, sim.moduleName)# use this file for reg too
    # Only biophys and regulation
    sim.init(biophys, regul, None, None)

    sim.addCell(cellType=0, pos=(0,0,0))

    # Add some objects to draw the models
    therenderer = Renderers.GLBacteriumRenderer(sim)
    sim.addRenderer(therenderer)
    sim.pickleSteps = 10

def init(cell):
    cell.targetVol = 3.5 + random.uniform(0.0,0.5)
    cell.growthRate = 1.0
    cell.n_a = 1
    cell.n_b = 1

def init(cell):

    cell.targetVol = 3.5 + random.uniform(0.0,0.5)

    cell.growthRate = 1.0

    cell.n = [1,1,1,1]                ## 1 copy of 4 different plasmids

def update(cells):
    for (id, cell) in cells.iteritems():

        r = 0.3+cell.n[0]/3.0

        g = 0.3+cell.n[1]/3.0

        b = 0.3+cell.n[2]/3.0

        y = 0.3+cell.n[3]/3.0

        r += y

        g += y

        cell.color = [r,g,b]

        if cell.volume > cell.targetVol:

```

```

cell.divideFlag = True

def divide(parent, d1, d2):
    d1.targetVol = 3.5 + random.uniform(0.0,0.5)
    d2.targetVol = 3.5 + random.uniform(0.0,0.5)
    plasmids = []
    for i in range(4):
        plasmids += [i]*parent.n[i]*2
        d1.n[i] = 0
        d2.n[i] = 0
    random.shuffle(plasmids)
    for p in plasmids[:4]:
        d1.n[p] += 1
    for p in plasmids[4:8]:
        d2.n[p] += 1

```

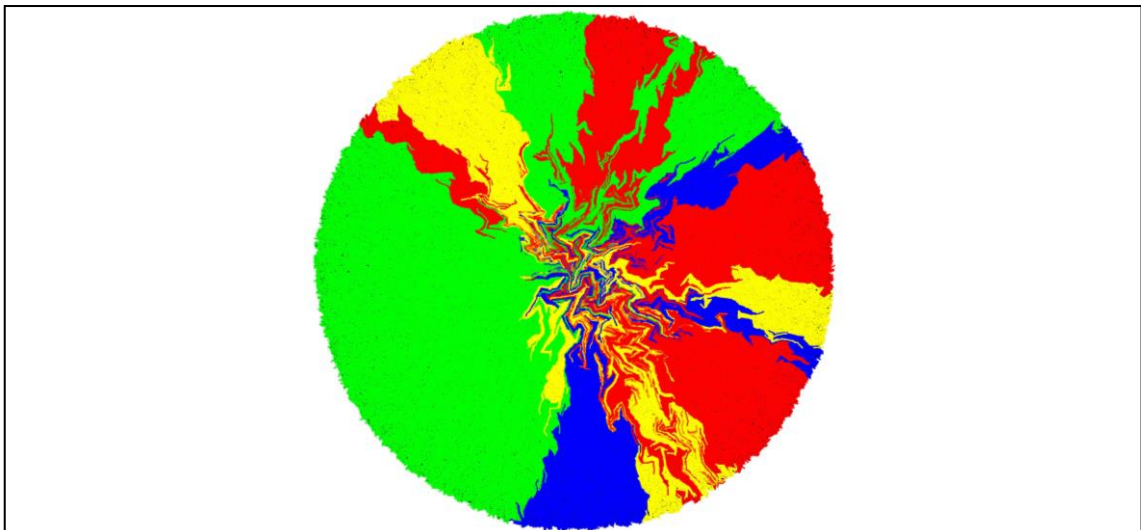


Figura A- 3: Ruptura de simetría a partir de célula fundadora con cuatro plásmidos distintos en su interior. La imagen muestra el estado final de una simulación efectuada con el código 3.

Código 4: Ruptura de simetría y regulación de la tasa de crecimiento celular de manera dominio específica.

```

import random
from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
from CellModeller.GUI import Renderers
from CellModeller.Integration.CLCrankNicIntegrator import CLCrankNicIntegrator #add
import numpy
import math

#Specify parameter for solving diffusion dynamics
grid_dim = (64, 8, 12) # dimension of diffusion space, unit = number of grid
grid_size = (4, 4, 4) # grid size
grid_orig = (-128, -14, -8) # where to place the diffusion space onto simulation space

def setup(sim):
    # Set biophysics, signalling, and regulation models
    biophys = CLBacterium(sim, jitter_z=False, max_cells=30000,max_sqs=192**2)

    regul = ModuleRegulator(sim, sim.moduleName)# use this file for reg too
    # Only biophys and regulation
    sim.init(biophys, regul, None, None)

    sim.addCell(cellType=0, pos=(0,0,0))

    # Add some objects to draw the models
    therenderer = Renderers.GLBacteriumRenderer(sim)
    sim.addRenderer(therenderer)
    sim.pickleSteps = 10

def numSpecies():
    return(0)

def numSignals():
    return(0)

def init(cell):
    cell.targetVol = 3.5 + random.uniform(0.0,0.5)
    cell.growthRate = 1.0
    cell.n_a = 1
    cell.n_b = 1

def update(cells):
    for (id, cell) in cells.iteritems():
        gr1 = 1.0
        gr2 = 0.5 # grow rate plasmid 2 (p2)
        cell.growthRate = (gr2-gr1)*cell.n_a*0.5 + gr1 #change growth rate linearly with p2

        cell.color = [0.1, cell.n_a/2.0, cell.n_b/2.0]
        if cell.volume > cell.targetVol:
            cell.divideFlag = True

def divide(parent, d1, d2):
    d1.targetVol = 3.5 + random.uniform(0.0,0.5)
    d2.targetVol = 3.5 + random.uniform(0.0,0.5)
    plasmids = [0]*parent.n_a*2 + [1]*parent.n_b*2
    random.shuffle(plasmids)
    d1.n_a = 0
    d1.n_b = 0
    d2.n_a = 0
    d2.n_b = 0
    for p in plasmids[:2]:

```

```

        if p == 0: d1.n_a +=1
        else: d1.n_b +=1
    for p in plasmids[2:4]:
        if p == 0: d2.n_a +=1
        else: d2.n_b +=1
    assert parent.n_a + parent.n_b == 2
    assert d1.n_a + d1.n_b == 2
    assert d2.n_a + d2.n_b == 2
    assert parent.n_a*2 == d1.n_a+d2.n_a
    assert parent.n_b*2 == d1.n_b+d2.n_b
    assert parent.n_a > 0 or (d1.n_a == 0 and d2.n_a == 0)
    assert parent.n_b > 0 or (d1.n_b == 0 and d2.n_b == 0)

```

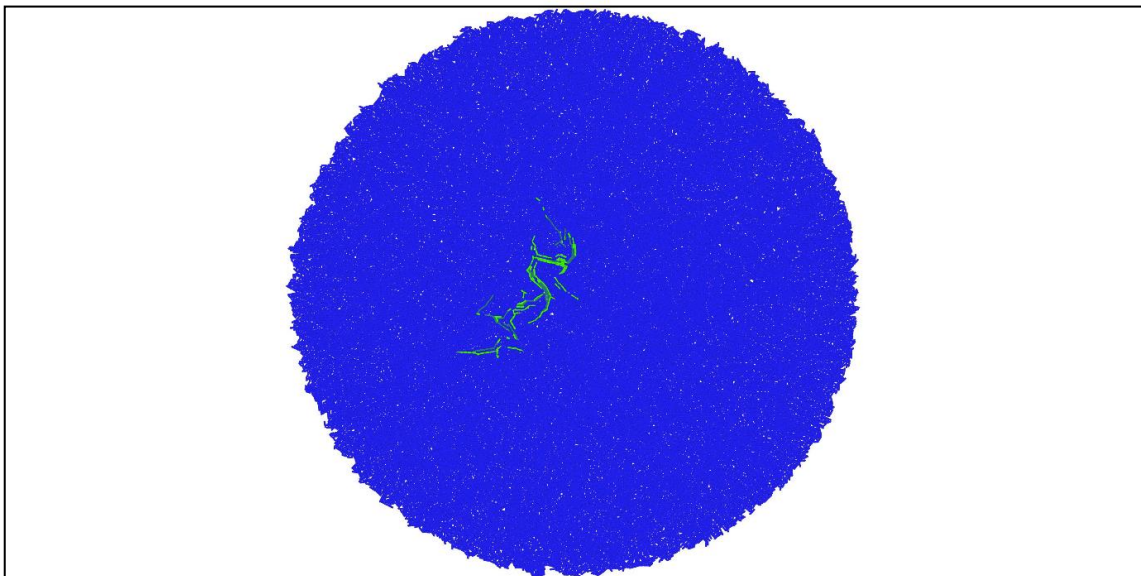


Figura A- 4: Ruptura de simetría y regulación del crecimiento celular de manera dominio específica desde el inicio. La imagen muestra el estado final de una simulación efectuada con el código 4, la cual comienza de una célula fundadora con dos plásmidos distintos en su interior y regula el crecimiento en cada célula (según los plásmidos que contenga) desde el comienzo (*i.e.* número de células =1).

Código 5: Ruptura de simetría y regulación de la tasa de crecimiento celular de manera dominio específica con control temporal.

El control temporal está representado como el momento en que se alcanza un número determinado de células en la colonia. Esto se implementó como una sentencia “if”, y solo al superar el valor indicado en su condición (en rojo), se modifica la tasa de crecimiento celular.

```
import random
from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
from CellModeller.GUI import Renderers
from CellModeller.Integration.CLCrankNicIntegrator import CLCrankNicIntegrator #add
import numpy
import math

#Specify parameter for solving diffusion dynamics
grid_dim = (64, 8, 12) # dimension of diffusion space, unit = number of grid
grid_size = (4, 4, 4) # grid size
grid_orig = (-128, -14, -8) # where to place the diffusion space onto simulation space

def setup(sim):
    # Set biophysics, signalling, and regulation models
    biophys = CLBacterium(sim, jitter_z=False, max_cells=30000,max_sqs=192**2)

    regul = ModuleRegulator(sim, sim.moduleName)# use this file for reg too
    # Only biophys and regulation
    sim.init(biophys, regul, None, None)

    sim.addCell(cellType=0, pos=(0,0,0))

    # Add some objects to draw the models
    therenderer = Renderers.GLBacteriumRenderer(sim)
    sim.addRenderer(therenderer)
    sim.pickleSteps = 10

def numSpecies():
    return(0)

def numSignals():
    return(0)

def init(cell):
    cell.targetVol = 3.5 + random.uniform(0.0,0.5)
    cell.growthRate = 1.0
    cell.n_a = 1
    cell.n_b = 1

def update(cells):
    for (id, cell) in cells.iteritems():
        if len(cells)>1000: # check if the number of cells surpass a defined
threshold
            gr1 = 1.0
            gr2 = 0.5
            cell.growthRate = (gr2-gr1)*cell.n_a*0.5 + gr1

            cell.color = [0.1, cell.n_a/2.0, cell.n_b/2.0]
            if cell.volume > cell.targetVol:
                cell.divideFlag = True
```

```

def divide(parent, d1, d2):
    d1.targetVol = 3.5 + random.uniform(0.0,0.5)
    d2.targetVol = 3.5 + random.uniform(0.0,0.5)
    plasmids = [0]*parent.n_a*2 + [1]*parent.n_b*2
    random.shuffle(plasmids)
    d1.n_a = 0
    d1.n_b = 0
    d2.n_a = 0
    d2.n_b = 0
    for p in plasmids[:2]:
        if p == 0: d1.n_a +=1
        else: d1.n_b +=1
    for p in plasmids[2:4]:
        if p == 0: d2.n_a +=1
        else: d2.n_b +=1
    assert parent.n_a + parent.n_b == 2
    assert d1.n_a + d1.n_b == 2
    assert d2.n_a + d2.n_b == 2
    assert parent.n_a*2 == d1.n_a+d2.n_a
    assert parent.n_b*2 == d1.n_b+d2.n_b
    assert parent.n_a > 0 or (d1.n_a == 0 and d2.n_a == 0)
    assert parent.n_b > 0 or (d1.n_b == 0 and d2.n_b == 0)

```

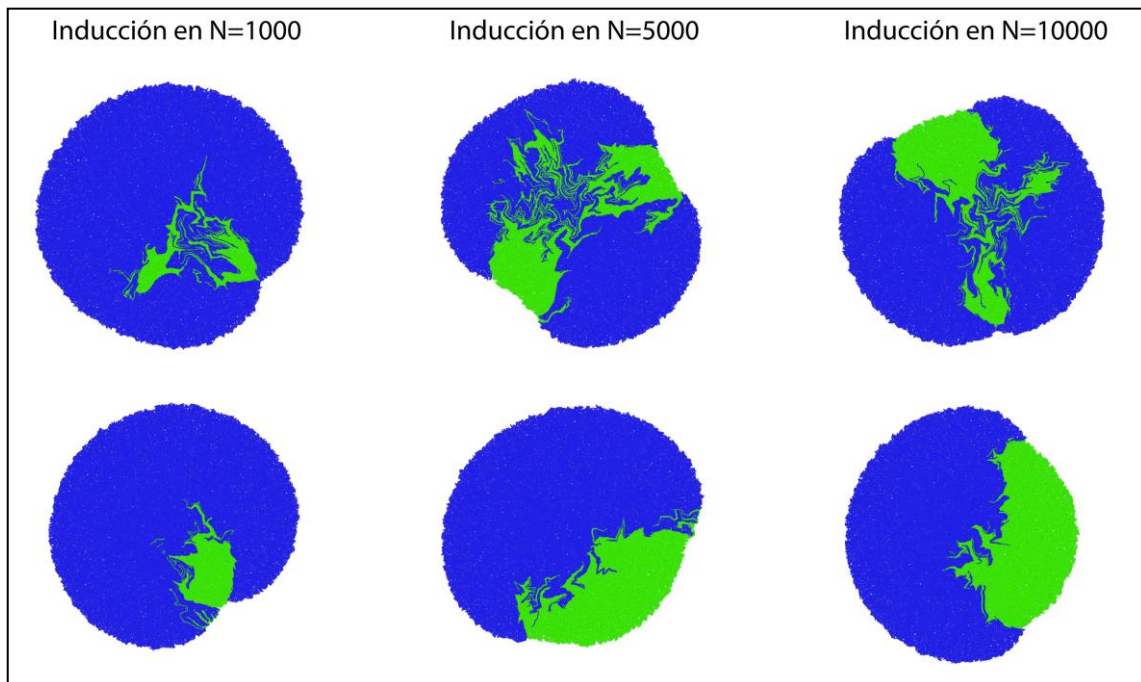


Figura A- 5: Ruptura de simetría y regulación del crecimiento celular de manera dominio específica con control temporal. La imagen muestra el estado final simulaciones efectuadas con el código 5, la cual comienza de una célula fundadora con dos plásmidos distintos en su interior y regula el crecimiento en cada célula desde que se supera un número umbral (N) de unidades en la colonia. Se muestran ejemplos de colonias bipartitas y multisector para umbrales de 1000, 5000 y 10000 células.

ANEXO D: PRIMERS UTILIZADOS

Se utilizaron los siguientes *primers* para realizar ensamblaje mediante el sistema UNSeq:

U1F: CATTACTCGCATCCATTCTCAGGCTG

U2R: GCTTGGATTCTGCGTTTGTTCGTC

U2F: GCTGGGAGTTCGTAGACGGAAACAAAC

U3R: CGACCTTGATGTTTCCAGTGCGATTG

U3F: GCACTGAAGGTCCTCAATCGCAC

U4R: GACTTTGCGTGTTGTCTTACTATTGCTGG

U4F: CTGACCTCCTGCCAGCAATAGTAAG

UXR: GGTGGAAGGGCTCGGAGTTGTGG

Los plásmidos segregantes (SEG) fueron amplificados mediante los *primers*:

1XCF: GTCCTGTCTGTGACAAATTGC

U1R: GAGACGAGACGAGACAGCCTGAG

UXF: CCAGGATACATAGATTACCACAACCTCCG

1XCR: GAGGGCAATTTGTCACAGGGTTAAG

Los vectores pAct fueron amplificados mediante los *primers*:

KAGF:TAATTACtaGTCCTTTTCCcgGAGaTcTGGGTATCTGTAAATTCTGCTAG
ACC

U1R: GAGACGAGACGAGACAGCCTGAG

UXF: CCAGGATACATAGATTACCACAACCTCCG

KAGR:TACCCAgAtCTCccgGAAAAAGGACtaGTAATTATCATTGACTAGCCCAT
CTCAATTG

Los *primers* utilizados para ensamblajes específicos por Gibson fueron:

A0CDR1: CCAACGATAATCGCACGTTTCATctagtatttctcttttcttagagtgtctag

A0CDF1: aaagaggagaaatactagATGAAACGTGCGATTATCGTTGGTGGC

A0CDR2: tgaaactgacgaacctgcagtcggttatcaTCAGCGCTGCAAAGCATAACGC

A0CDF2: ATGCTTTGCAGCGCTGAtgataacgcactgcaggttcgtcag

VioABER:GTGGTGGCTCACGGTTCTCCATTTTTTTTACCTCCTTAAGGATCCTT
AGGCCTC

VioABEF:AAGGATCCTTAAGGAGGTAAAAAAAATGGAGAACCGTGAGCCAC
CACTG

1XKR2: tcaTATAAACGCAGAAAGGCCACCCG

1XKF2: CGGGTGGGCCTTTCTGCGTTTATAtg

U1U2R2: gaggagaaataactagATGAGTGTGATCGC

U1U2F2: GCGATCACACTCATctagtatttctcctc

U2U3R: tgcTCTAGAagcggccgcGAATTCCGGCTTGGATTCTGCGTTTGTTTCC

U2U3F: ACGCAGAATCCAAGCCGGAATTCgcgccgctTCTAGAgc

T7TerR1: tggcagcagccaactcagcGAATTCTTATTTATACAGTTCGTCCATACCGTG

T7TerF1: GGACGAAGTGTATAAATAAGAATTCgctgagttggctgctgcc

T7TerR2: AACCCTTAGTGACTCGAATTCgcttaatgcgccgctacagggc

T7TerF2: gccctgtagcggcgccattaagcGAATTCGAGTCACTAAGGGTTAGTTAGTTAG

2C3R2: gtgcgattgaggaccttcagtgcccgctactagtataaacgcagaaaggc

2C3F2: tctgcgtttatatactagtagcgggcactgaaggtcctcaatcgactgg

19stR: taggactgagctagctgtaaactgtgggtccgaattccg

19stF: cggaattcggaaccacaagtttacagctagctcagtcctaggtattatgctagc

219R: CTTGTGGGTCCGAATTCGGCTTGGATTCTGCGTTTGTTTCCG

219F: AAACGCAGAATCCAAGCCGGAATTCGGACCCACAAG

2iiidR73R: GATGTTAATCGTGTTTCATctagtactttcctgtgtgatctagtgctc

2iiidR73F:

actagatcacacaggaaagtactagATGAACACGATTAACATCGCTAAGAACGAC

SEG6R: tgcagcggccgctactagtaTATAAACGCAGAAAGGCCACCCG

SEG6F: GTGGGCCTTTCTGCGTTTATAtactagtagcggccgctgcagg

SEG9R: taatacctaggactgagctagctgtaaactGTTAGTTAGCCCTTAGTGACTCGAATTCG

SEG9F: tacagctagctcagtcctaggtattatgctagctactagagaaaggagaaatactag

SEG8F:tttacagctagctcagtcctaggtattatgctagcTACTAGAGAAAGAGGAGAAATACTAG
atgagcaaaggtgaagaactgttcacc

SEG12R1: GTCGGCAAATAAGAGCTCGCTTGGtactagctccggcaaaaaaacgggc

SEG12F1: ccgttttttgccggagctagtaCCAAGCGAGCTCTTATTTGCCGACTAC

SEG12R2: ctagcataatacctaggactgagctagctgtaaactTCCTCTAGAGTCGACCTGCAGGC

SEG13R1: ATGCTCGATGAGTTTTTCTAATAATACTAGCtccggcaaaaaaacgggc

SEG13F1: ccttgcccgttttttgccggaGCTAGTATTATTAGAAAAACTCATCGAGCATC

SEG13R2:

agcataatacctaggactgagctagctgtaaactACTAGAGCTGATCCTTCAACTCAGC