UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Design and implementation of digital twins of synthetic inducible gene expression circuits

End of Degree Project

Bachelor's Degree in Biomedical Engineering

AUTHOR: Pushkareva , Anna

Tutor: Picó Marco, Jesús Andrés

Cotutor: Boada Acosta, Yadira Fernanda

ACADEMIC YEAR: 2022/2023

Agradecimientos

Me gustaría dedicar este trabajo a todas aquellas personas que me apoyaron y estuvieron a mi lado a lo largo de esta etapa.

En primer lugar, a Alejandro, Yadira y Jesús, que me acogisteis y me enseñasteis todo lo que sé ahora de biología sintética y trabajo de laboratorio. Os estoy infinitamente agradecida por haber tenido siempre tanta paciencia conmigo y por haberme hecho sentir parte de vuestra familia. Debo hacer mención especial también a Andrés, con quien siempre es un placer encontrarme en el laboratorio y compartir unas risas.

A Caki, quien ha sido un gran apoyo a lo largo de estos cuatro años, y con quien he compartido tanto risas como lágrimas. Eres la mejor amiga y compañera que me ha podido dar Valencia, y es inmesurable todo el cariño que te tengo.

A mis amigos de *Estamos burgir*. Carolina, Paula, María, Alba, Raquel, Arnau, Pablo y Jorge. Sois los mejores compañeros de carrera que podría haber tenido, y son incontables todas las veces que me he reído hasta llorar con vosotros a lo largo de este tiempo.

A Sofi, mi mejor amiga, que a pesar de estar lejos, sé que siempre te puedo llamar y parecerá que estamos justo al lado.

A los *Qkis*, David, Juanlu, Rori, Mumi, Kari y Pachi. Gracias por estar ahí cuando más lo he necesitado.

A Alejandro. Gracias por haber estado a mi lado durante estos últimos meses, por haberme apoyado y por haber sido mi *proof-reader*.

A mi familia. A mi mamá Victoria, a mi hermanito Artemiy, a mi papá Sergey y a mi abuelita Galya. Gracias por haberme ayudado a convertirme en la persona que soy hoy.

Abstract

Synthetic Biology allows the construction of genetic circuits with a desired behaviour by modifying and combining different genetic elements that exist in nature. The characterization of these circuits, taking into account their genetic parts and the microorganism where they are implemented, plays an essential role when trying to build genetic circuits with more complex behaviour.

The design and implementation of digital twins of genetic circuits allows their characterization taking into account mechanistic and systems engineering principles. Each digital twin represents by means of a dynamic model the temporal behaviour *in silico* of a genetic circuit built *in vivo* in the laboratory.

Therefore, in the present project, the characterization of the basic parts of inducible genetic circuits built in the laboratory is sought. To this end, semi-empirical, partly first-principles based, dynamic model structures for these parts are devised. Then the experimental identification of its associated parameters, and the validation of experimental data for each bio-part with the results obtained from computational simulations of their corresponding digital twins is carried out.

The characterization was developed in a systematic way following the Design-Build-Test-Learn (DBTL) synthetic biology cycle. Ten genetic circuits were designed and constructed that combine different DNA sequences (promoter, ribosome binding site, and copy number) that produce a fluorescent protein as an indicator. This protein can later be exchanged for any other protein of practical interest.

With these ten circuits, several experiments were performed to analyse how their fluorescence level changes when an activation signal (Acyl-homoserine lactones, AHL) is induced. The genetic circuits were implemented in *Escherichia coli* bacteria as self-replicating biofactories that work 24 hours a day, 365 days a year. The genetic circuits require optimal functioning, which in turn demands more accurate models to meet the optimal cell and circuit demands.

In the learning phase, the estimation of the parameters of these models was carried out, minimising the difference between the experimental data of the measured fluorescent protein and that calculated by the dynamic model (the digital twin) designed for each of the genetic circuits. The estimation of the parameters allowed us to explain up to approximately 80% of the behaviour

observed in the laboratory, which is a good initial approximation considering that we are working with living systems.

Through the process of characterising parts of a genetic circuit, a deeper understanding of the behaviour and performance of individual parts and in combination with other parts can be obtained. With this knowledge, researchers can save valuable time in designing new and more sophisticated genetic devices that have a reproducible and reliable desired behaviour in any biotechnological application.

Keywords: Dynamic modeling, digital twin, parameter estimation, synthetic biology, experiment design.

Resumen

La Biología Sintética permite la construcción de circuitos genéticos con un comportamiento deseado, modificando y combinando diferentes elementos genéticos que existen en la naturaleza. La caracterización de estos circuitos teniendo en cuenta sus partes genéticas y el microorganismo donde se implementan, juega un papel esencial cuando se intenta construir circuitos genéticos con comportamientos más complejos.

El diseño e implementación de gemelos digitales de circuitos genéticos permite su caracterización teniendo en cuenta principios mecaninicistas y de ingeniería de sistemas. Cada gemelo digital representa mediante un modelo dinámico el comportamiento temporal *in silico* de un circuito genético construido *in vivo* dentro del laboratorio.

En el presente proyecto se busca la caracterización de las partes básicas de circuitos genéticos inducibles construidos en el laboratorio, a través de el planteamiento de estructuras adecuadas de modelos dinámicos semi-empíricos parcialmente basados en primeros principios biológicos. Posteriormente se lleva a cabo la identificación experimental de los parámetros asociados, y la validación de datos experimentales de cada parte con los resultados obtenidos de las simulaciones computacionales de sus correspondientes gemelos digitales.

La caracterización se desarrolló de forma sistemática siguiendo el ciclo de la Biología Sintética Diseño-Construcción-Prueba-Aprendizaje (DBTL). Se diseñaron y construyeron diez circuitos genéticos que combinan distintas secuencias de ADN (promotor, lugar de unión de ribosomas, y número de copias) que producen una proteína fluorescente a manera de indicador. Esta proteína puede ser intercambiada a posteriori por otra de interés práctico.

Con estos diez circuitos se realizaron experimentos para analizar cómo cambia el nivel de fluorescencia cuando se induce una señal de activación (moléculas de comunicación entre bacterias Acyl-homoserine lactones AHL). Los circuitos genéticos fueron implementados en bacterias *Escherichia coli* como biofábricas que se auto-replican y trabajan 24 horas, los 365 días del año. Para su desempeño adecuado en el contexto celular, los circuitos genéticos requieren un funcionamiento óptimo, lo que a su vez demanda modelos más exactos que permitan adecuar las condiciones óptimas de desempeño.

En la fase de aprendizaje se llevó a cabo la estimación de los parámetros de los modelos, minimizando la diferencia entre los datos experimentales de la proteína fluorescente medida, y la

calculada por el modelo (*gemelo*) digital diseñado para cada uno de los circuitos genéticos. La estimación de los parámetros permitió explicar aproximadamente el 80% del comportamiento observado en el laboratorio, lo que supone una buena aproximación inicial teniendo en cuenta que estamos trabajando con sistemas vivos.

Gracias al proceso de caracterización de partes de un circuito genético se puede obtener una comprensión más profunda del comportamiento y el desempeño de las partes individuales y en combinación con otras.Este conocimiento puede ahorrar un tiempo valioso en el diseño de nuevos y más sofisticados dispositivos genéticos que tengan un comportamiento deseado reproducible y fidedigno en cualquier aplicación biotecnológica.

Palabras Clave: Modelado dinámico, gemelo digital, estimación de parámetros, biología sintética, diseño de experimentos.

Resum

La Biologia Sintètica permet la construcció de circuits genètics amb un comportament desitjat, modificant i combinant diferents elements genètics que existeixen a la natura. La caracterització d'aquests circuits tenint en compte les seves parts genètiques i el microorganisme en el qual s'implementen, juga un paper essencial quan s'intenta construir circuits genètics amb comportaments més complexos.

El disseny i implementació de bessons digitals de circuits genètics permet la seva caracterització tenint en compte principis mecanicistes i d'enginyeria de sistemes. Cada bessó digital representa mitjançant un model dinàmic un comportament temporal *in silico* d'un circuit genètic construït *in vivo* al laboratori.

En aquest projecte se cerca la caracterització de les parts bàsiques de circuits genètics induïbles construïts al laboratori, a través del plantejament d'estructures adequades de models dinàmics semi-empírics parcialment basats en primers principis biològics. Posteriorment es duu a terme la identificació experimental dels paràmetres associats, i la validació de dades experimentals de cada part amb els resultats obtinguts de les simulacions computacionals dels seus corresponents bessons digitals.

La caracterització es va desenvolupar de forma sistemàtica seguint el cicle de la Biologia Sintètica Disseny-Construcció-Prova-Aprenentatge (DBTL). Es dissenyaren i construïren deu circuits genètics que combinen diferents seqüències d'ADN (promotor, lloc d'unió de ribosomes, i número de còpies) que produeixen una proteïna fluorescent a manera d'indicador. Aquesta proteïna pot ser intercanviada a posteriori per altra d'interès pràctic.

Amb aquests deu circuïts es realitzaren experiments per analitzar com canvia el nivell de fluorescència quan s'indueix una senyal d'activació (molècules de comunicació entre bactèries, Acyl-homoserine lactones o AHL). Els circuïts genètics foren implementats en bactèries *Escherichia coli* com biofàbriques que s'auto-repliquen i treballen 24 hores, els 365 dies de l'any. Pel seu adequat desenvolupament en el context cel·lular, els circuits genètics requereixen un funcionament òptim, el qual a la vegada demana models més exactes que permetin adequar les condicions òptimes de desenvolupament.

En la fase d'aprenentatge es va dur a terme la estimació dels paràmetres dels models, minimitzant la diferència entre dades experimentals de la proteïna fluorescent mesurada, i calculada pel model

(*bessó*) digital dissenyat per a cada un dels circuits genètics. La estimació dels paràmetres va permetre explicar aproximadament el 80% del comportament observat en el laboratori, el qual suposa una bona aproximació inicial tenint en compte que estam treballant amb sistemes vius.

Gràcies al procés de caracterització de parts d'un circuit genètic se pot obtenir una comprensió més profunda del comportament i desenvolupament de les parts individuals i en combinació amb altres. Aquest coneixement pot estalviar un temps valuós en el disseny de nous i més sofisticats dispositius genètics que tinguin un comportament desitjat reproduïble i fidedigne en qualsevol aplicació biotecnològica.

Paraules Clau: Modelatge dinàmic, bessó digital, estimació de paràmetres, biologia sintètica, disseny d'experiments.

List of Figures

List of Tables

Contents

Part I

DISSERTATION REPORT

Chapter 1

Introduction

1.1 Background

Synthetic biology is a rapidly growing interdisciplinary field that combines principles from both biology and engineering. At its core, the aim is to engineer living cells and organisms to perform specific functions or carry out a desired behaviour that may not exist in nature. To achieve this, it is possible to introduce new genetic material, modify the existing one, or even design new biological systems from scratch.

The term *synthetic biology* was used for the first time by Barbara Hobom, referring to genetically modified bacteria through recombinant DNA technology (Hobom, 1980). In this sense, it is said that it is *biology* because it uses biological systems and *synthetic* because, after all, they are altered through human intervention.

Synthetic biology applications are broad, ranging from medicine, food and agriculture, climate change, and manufacturing (Benner and Sismour, 2005). For example, the manufacture of new drugs or vaccines, the application of new therapies to improve health and well-being, the development of new methods for environmental monitoring and pollution remediation, and even the design of novel biosensors in medicine.

In recent years, the field of synthetic biology has made significant strides in understanding and engineering gene expression pathways. By engineering gene expression, it is possible to manipulate the timing, magnitude, and duration of a specific gene's activity, which can profoundly affect the cell's function. This has led to the development of synthetic gene circuits, which are networks of genes designed to perform specific functions in living cells and can be used for various applications, such as the ones mentioned above (Xie and Fussenegger, 2018).

The element that distinguishes synthetic biology from traditional molecular and cellular biology is the focus on the design and construction of core components (e.g., DNA gene sequences, proteins properties, genetic circuits, or metabolic pathways) that can be modelled, understood, and tuned to meet specific performance criteria.

This is known as the DBTL cycle of synthetic biology (Design, Build, Test, and Learn) that enables scientists to assemble these smaller parts and devices into larger integrated systems to solve specific problems. Ultimately, albeit it relies strongly on biology, it can not be forgotten that it is also an engineering field. Therefore, it must also be approached as such, and just like traditional engineering disciplines, synthetic biology relies on the systematic application of mathematical modelling, computer simulations, and experimental techniques to engineering living organisms with desired functions and traits (Cheng and Lu, 2012).

1.2 Motivation

Just like electrical circuits, genetic circuits in the realm of synthetic biology are engineered by combining different biological components called **BioBricks** or **bioparts** (mainly: promoters, coding sequences, and terminators, which will be further explained in Section 3), which lead to the possibility of controlling and manipulating cellular processes in a wide range of ways (Brown, 2007). Depending on the used parts, the function of a circuit can change drastically, varying from producing biofuels and biodegradable materials to pharmaceutic components – just like the design of machines that perform one task or another depending on the elements that comprise it.

Nevertheless, just like happens with, for example, electrical resistances, BioBricks have their specifics and responses depending on their type, DNA sequence, interaction with other bioparts, and so on. Therefore, to construct circuits successfully, it is crucial to know how these parts work, and thus their characterization is needed so their mathematical modelling can be as accurate as possible before being developed in a laboratory.

Understanding just how these bioparts work is especially relevant in the current state of synthetic biology. Despite the rapid progress in this field, the existing libraries of BioBricks are still under development, and many parts have not undergone thorough analysis. This lack of comprehensive understanding poses a challenge for researchers in designing new genetic circuits efficiently.

However, thanks to characterization, it is possible to gain a deeper understanding of the behaviour and performance of individual bioparts on a quantitative basis (Hirst, 2014). With this knowledge, researchers can save valuable time by skipping unnecessary iterations and focusing on identifying the optimal combination of parts to achieve the desired output.

1.3 Summary

The **characterization** process involves studying the BioBricks experimentally under different conditions and obtaining a series of performance metrics, which will be then used to adjust the parameters of the mathematical model representing the biopart(s) of interest. For example, a possible metric could be the production rate of a specific protein or the response to external stimuli.

In this project, the main biopart of interest is the **inducible promoter Plux**. The promoter regulates the expression rate of a particular gene (thus, protein expression) – in this case, since it is inducible, this regulation is dependant on an external molecule that acts as an activator-type Transcription factor (TF), the N-Acyl homoserine lactone (AHL).

A total of **five Ribosome binding sites** of varying strengths will also be studied in parallel to the promoter. Specifically, each of the RBSs will be combined in a single synthetic circuit with the promoter to analyze the effect of changing the strengths and their interaction with the promoter. Additionally, the Copy Numbers (C_N) of two destination vectors will be evaluated too.

To achieve this, the first step will be the mathematical modelling of a genetic circuit in which a constitutive and an inducible part are combined (these concepts will be further developed in Chapter 3) using ordinary differential equations (ODEs). This initial prediction will help understand the behaviour of the actual circuit constructed in the laboratory.

Once the mathematical modeling is completed, the five circuits will be constructed using molecular and synthetic biology techniques. The correct synthesis of these circuits will be verified using techniques such as colony PCR. Furthermore, these circuits will be placed in two different destination vectors (see Chapter 3), resulting in two variations for each circuit, totalling ten constructions.

After the assembly phase, these circuits will be incorporated into *E. coli* cells through cell transformation techniques to generate bacterial cultures, and a series of experiments will be conducted to collect experimental data. These experiments will involve a 16-hour growth process, during which the molecule N-Acyl homoserine lactone (AHL) will be added to the cultures at different concentrations during the first hour to induce the production of Green fluorescent protein (GFP). This will allow studying the effect of different concentrations in combination with the changes of biological parts among constructions using various metrics, providing insights into the behaviour of the studied circuit.

The intention behind conducting these experiments is to extrapolate the obtained laboratory data to fine-tune the mathematical model. The ultimate goal is to adjust the parameters as accurately as possible, creating a digital twin of the assembled biological parts. The central relevance of such twins lies in their ability to provide accurate predictions of the outputs when specific inputs are introduced into the system. This enables scientists to harness the potential offered by the constructed circuits fully.

In summary, this project involves a combination of experimental and computational work. Beginning with an initial mathematical model of a circuit, it is then tested in the laboratory, and based on the obtained results, the model is refined to increase its resemblance to the experimental data as much as possible.

Chapter 2

Objectives

2.1 General objectives

The general objectives of this thesis are the following:

1. Characterization of gene circuits. This allows to deeply understand the behaviour and performance of individual bioparts on a quantitative basis in the laboratory.
2. Use the Design, Build, Test, and Learn cycle to achieve the successful design of a biological system. This enables the design and implementation of core components (e.g., DNA parts, gene circuits, metabolic pathways) that can be modeled, understood and tuned to meet specific performance criteria.
3. Design, assemble and test basic DNA parts. Also, incorporate them into larger integrated biological systems to solve a specific problem.
4. Design a digital twin or mathematic model of gene circuits that will represent their in vivo behaviour by applying engineering principles.
5. Identify the digital twin's parameters by means of experimental data and optimization techniques. These parameters will serve as key inputs to mathematical models that represent said circuits.
6. Simulate and predict the system's behaviour under different conditions. Compare the digital twin's performance to the one observed in the laboratory experimentally.

2.2 Specific objectives

The specific objective is to characterize some of the basic parts of a gene circuit by estimating the values of the following parameters:

- **Inducible promoter Plux.** This will include three main parameters:
  - Half-concentration (k_{dlux}). This is the AHL concentration at which half of the promoter's binding sites are occupied.
  - Cooperativity coefficient (n). It is an estimator of the affinity between the promoter and the Transcription factor.
  - Basal expression (α) for high-copy and low-copy circuits. It represents how much GFP is produced when there is no induction with AHL.
- **Five Ribosome binding sites (RBS).** The binding site is the mRNA region to which ribosomes bind to begin the translation process. Specifically, the focus will be set on determining their strength, which is the affinity between the ribosome and the site.
  - RBS B0030
  - RBS B0031
  - RBS B0034
  - RBS J61100
  - RBS J61101
- **Two Copy Numbers (C_N).** This number is an estimator of the amount of plasmids contained in a single cell.
  - High-copy vector pRMS_Omega1
  - Low-copy vector pIASO1S471 (Alarcón López, 2019)

In order to do this, the following specific goals were set:

1. Design five genetic circuits, one per each combination of Plux promoter - RBS pair.
2. Develop an initial mathematical model to predict the behaviour of said circuits.
3. Build the five genetic circuits and incorporate each of them in two different backbones (high-copy and low-copy).
4. Test the circuits in a laboratory environment through experiments with AHL induction and gather experimental data.
5. Adjust the parameters of the mathematical model to fit the obtained data.
6. Characterize the biological elements with the obtained adjusted parameters.

Part II

PRELIMINARIES

Chapter 3

State of the Art

3.1 Gene expression

Protein production or gene expression can be defined as *the process by which the information encoded in a gene is turned into a function* ("Synthetic Biology", 2022). This can be manifested, for example, through protein production.

The **Central Dogma of Molecular Biology**, proposed by Francis Crick in 1970, comprises the flow of genetic information within a biological system. According to the Central Dogma, genetic information is transferred via **transcription** from Deoxyribonucleic acid (DNA) to Messenger ribonucleic acid (mRNA), and then via **translation** from mRNA to protein. In simple terms, this means that the information is stored in the DNA and has to be retrieved through the two mentioned processes to be interpreted in the form of a function, which is the protein encoded in its sequence (see Figure 3.1).

Transcription starts with the binding (or union) of the RNA polymerase (RNApol) to the Coding sequence (CDS), which is the DNA region called *promoter* to generate molecules of Messenger ribonucleic acid (mRNA). The mRNA carries protein information from a gene Coding sequence (CDS) that would be translated into a protein afterwards.

It is essential to mention that transcription can be either *constitutive* or *inducible*. Constitutive transcription occurs without requiring additional proteins to intervene in the process. Inducible transcription, on the other hand, requires the presence of additional proteins called Transcription factors (TF). Said factors can either activate or repress the transformation from DNA to mRNA.

The next step is **translation**, through which mRNA is decoded and translated into a protein. In this case, the binding of ribosomes, which function as the builders of the proteins, to the Ribosome binding site (RBS), results in the successive addition of peptides, which are the building blocks of proteins. These blocks make a polypeptidic chain that will then fold itself through a series of different chemical reactions, resulting in the end in a functional protein.

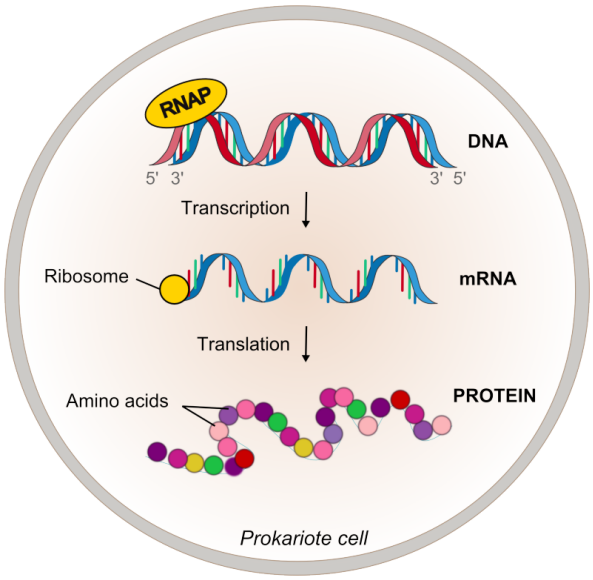


Figure 3.1: Description of the gene expression process. The RNApol transcribes the DNA into mRNA, and finally the RBS translates it into an amino acid sequence (the protein). (Figure taken from Boada Acosta, 2018)

3.2 Bioparts

To produce a protein inside a cell, there are four basic elements, which, as introduced in Chapter 1, are the BioBricks or bioparts.

1. **Promoter**

   The promoter plays a fundamental role in initiating the transcription process by recruiting the RNA polymerase (RNApol), and it can be constitutive or inducible. In the case of a **constitutive promoter**, the binding rate between the RNApol and the promoter remains constant, ensuring a continual transcription of a gene. Roughly, this means that the speed at which these two elements bind and transcript DNA does not change over time.

   However, in the case of an **inducible promoter**, the binding will depend on the presence of the Transcription factor, as well as its type. If this factor is an *activator*, its interaction with the promoter will favour the binding of the RNApol, so the transcription will be triggered. Conversely, if the factor is a *repressor*, this interaction will stop the RNApol from binding to the promoter, thus inhibiting the transcription process.

2. **Ribosome binding site (RBS)**

   Similar to the promoter, the RBS is a region in the mRNA where ribosomes are bound to start the translation process. It is relevant to point out that, depending on the RBS type, the binding affinity of ribosomes to RBS might be changed. This property is known as RBS strength. Therefore, the stronger the RBS, the more protein is expressed.

3. **Coding sequence (CDS)**

   The CDS is the DNA sequence that codes for the exact succession of amino acids or peptides that form the protein. Therefore, it is first transcribed from DNA to mRNA, and then translated from mRNA to a specific functional protein.

4. **Terminator**

   During transcription, the terminator is another DNA sequence that stops the process by disassociating the RNApol from the promoter.

When a promoter, RBS, CDS and terminator are in a vector, we have a basic Transcription Unit (TU) to produce a protein. However, this circuit by itself cannot be incorporated properly into the cell, so it must be done in the form of a plasmid – which is the union between the TU and a specific **vector**.

A vector or **backbone** serves as the 'carrier' of one or more of the previous parts. Just like the TU, it is a DNA sequence, from which two specific regions can be highlighted:

- **Origin of replication.** This is a region in a plasmid that defines the initial amount of DNA to be transcribed, and it can be mathematically represented by the Copy Number (C_N).

  This site also allows for the division of vectors in two main groups: high-copy and low-copy. As can be deduced by the name, in the first case the C_N will be higher.

- **Antibiotic resistance.** This site in the plasmid provides the cell and the gene circuit with resistance to a specific antibiotic. This way, it is possible to avoid contamination and ensure that only the cells that carry the desired gene circuit will survive.

Assembling together these different parts, or several transcriptional units, is what allows to implement a **genetic circuit**. In order to accomplish this, we will be set on the **Golden Braid (GB) assembly**, which was the method used to construct all the circuits.

3.3 The Golden Braid Assembly

Assembling DNA means to unite separate parts of it to construct a larger sequence with a particular function derived from the combination of said pieces. To accomplish this, specific enzymes are needed to 'cut' the DNA, called the **endonucleases** or **restriction enzymes**; then these DNA fragments need to be 'glued' together, by other enzymes called **ligases**.

The Golden Braid assembly is based on the type IIS restriction enzymes, which interact with the DNA in two different sites: the *recognition site*, to which they bind, and the *cleavage site*, where they cut the DNA (Szybalski et al., 1991; Sarrion-Perdigones et al., 2011).

Now, it is relevant to mention that using this assembling technique, it is possible to construct circuits with different levels of complexity. Inferior levels can be combined into higher ones to obtain more complex circuits.

- **Level 0** is the most basic level, where each part (promoter, RBS, CDS, terminator, and backbone) is separated in different plasmids. As mentioned above, it cannot be incorporated directly into the cell yet, since a vector is needed.

- The **Level 1** construction is the first assemble between Levels 0, for example to constitute a TU incorporated into a backbone. In this case, since only one circuit with one CDS is present, this level allows for the production of a single protein.

- The **Level 2** assembly, which results by joining two Level 1 constructs, makes it possible to trigger the production of two proteins, one per each CDS.

As pictured in Figure 3.2, for levels one and two the antibiotic resistance always remains the same. Level 1 constructs are always created using vectors that are resistant to kanamycin, and to spectinomycin in the case of Level 2. This allows for the correct selection of colonies once the assembly is conducted and cells are transformed (see the corresponding protocols in Section 4.4.2).

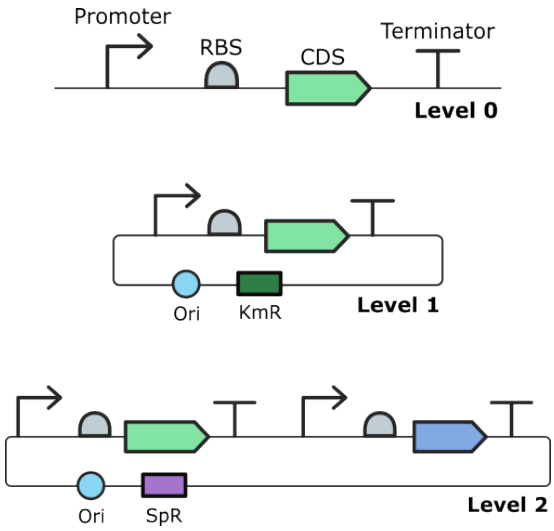


Figure 3.2: Level 0, 1, and 2 assemblies.

Finally, it is important to describe how each Level operates vector-wise. Since Level 0 is exclusively the TU, it does not require a vector to be constructed. However, Levels 1 and 2 do need backbones, and these are chosen depending on whether it is desired to combine them afterwards with other circuits.

This way, for Level 1 constructs, α-type vectors are needed, which can be subdivided into $\alpha 1$ or $\alpha 2$. This difference is necessary because if the intention is to further combine them into a Level 2 construction, the backbones necessarily need to be of two different sub-types. The same applies to ω-type vectors, although this escapes the scope of this project and thus will not be discussed further.

Additionally, the chosen vector will define the restriction enzymes to be used in the assembly. The **BsaI** is needed for the construction of assemblies using the type α backbone, and the **BsmBI** for Ω vectors. The ligase used is the T4, and in this case is independent from the levels.

3.4 Mathematical modeling of synthetic circuits

Any system can be described as a black box that produces a certain output under an external stimulus (Karplus, 1977). Therefore, this same description can be applied to biological systems – in this case, the cell or group of cells would be the black box, working together in a gene circuit that has a specific function.

For biological systems, mathematical models are important because they allow to explore diverse scenarios and combination of parts before creating a genetic circuit in the laboratory. This way, instead of conducting experiments and basing decisions on trial-and-error, it is possible to predict the behaviour of a gene circuit in a computational environment, which makes the design and building processes more efficient, cost-effective, and time-saving.

Dynamic models are particularly interesting in this context, since they allow us to analyze the evolution of a system in a customized temporal window.

In this context, two types of models exist: **stochastic** and **deterministic**. In the first case, the reactions are modeled taking into account the probability of their occurrence, therefore making them more realistic. However, this consideration often supposes a greater challenge in terms of solving the mathematical equations that compose the model, which translates into an increase in computational demand.

In deterministic models it is assumed that the concentration of the biological species that intervene in the system are large enough and vary continuously over time (Boada Acosta, 2018). For this reason, they are often easier to interpret and analyze – although they are still able to accurately describe the system.

In fact, there are many cases in which the use of deterministic models provides a very good approximation of the real-world behaviour of the studied system, such as modeling growth factor-induced angiogenesis (Sun et al., 2005) or restriction point control in animal cells (Novák and Tyson, 2004). Therefore, considering the potential offered by these types of models, their use has been chosen for this project.

3.5 Performance metrics

The goal of conducting the experiments is observing the response of the system in the form of different **performance metrics** to external stimuli and incorporating that information into the mathematical model. In this case, the external stimulus will be the addition of AHL to the bacterial culture, and the metrics obtained to evaluate the system's response will be **absorbance** and **fluorescence**.

3.5.1 Absorbance

Absorbance, or Optical Density (OD), is a commonly used parameter to indirectly quantify the amount of cells present in a bacterial culture, and is based on the physical principle of **light absorbance**. By means of a spectrophotometer, light of a specific wavelength (usually 600 nm) is passed through a sample and absorption is measured on the other side. Logically, the greater the concentration of bacteria is, the more light will be absorbed (Sutton, 2006).

More specifically, this metric is based on the Beer-Lambert law, which states, as shown in Equation 3.1, that the absorbance of a sample is proportional to its concentration (c), the optical path length or width of the cuvette where the sample is contained (l), and a constant ϵ, which is the molar absorption and depends on the analyzed compound (Clark, 2013).

$$A = \epsilon \cdot c \cdot l \tag{3.1}$$

While it is possible to calculate the concentration of a culture using the Beer-Lambert law, there is a scattering problem when its concentration increases to a certain level, causing Equation 3.1 to deviate from linearity and become parabolic in relation to the number of cells (Boada et al., 2019). That is why, although it is a good metric for initial normalization of cultures and tracking of their growth during experiments, when the intention is to estimate the approximate number of cells, it is more useful to calculate it in terms of particles.

Particles

To determine the equivalence between absorbance and particles, it is necessary to apply Equation 3.2 to the measured value (Boada et al., 2019).

$$\begin{aligned} &\text{particles} = p_0 \cdot (\text{Abs}_{600})^{p_1} \\ &\text{where} \quad p_0 = 10^{8.062} \quad \text{and} \quad p_1 = 1.185 \end{aligned} \tag{3.2}$$

Thus, the equivalence between OD and particles can be established as a linear relationship (see Figure 3.3), as long as the absorbance values are above 0.025 and slightly below 1.

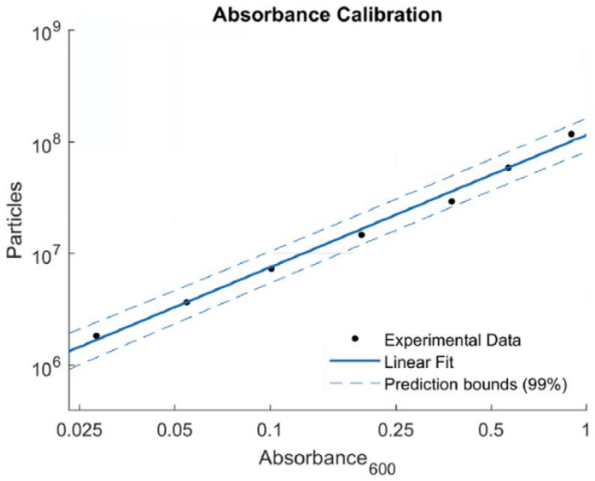


Figure 3.3: Equivalence between measured absorbance and the amount of particles (cells) in the studied medium. Graph taken from Boada et al., 2019.

3.5.2 Fluorescence

This parameter is based on the photoluminiscence phenomenon, in which a specific compound (the fluorophor) emits light after absorbing it ("Fluorescence Intensity Measurements | BMG LABTECH", n.d.). Since the wavelengths at which said light is absorbed and then emitted are different, it is possible to use this property for sample analysis.

Thus, fluorescence intensity can be measured by exciting the fluorophore at a specific wavelength and then measuring the light emitted at a different wavelength. Said intensity is usually defined in terms of Relative Fluorescent Units (RFU), and it presents a linear correlation with the concentration of fluorophore.

This is a useful parameter for gene expression quantification when the produced protein of interest presents fluorescence. Specifically, the Green fluorescent protein (GFP) is a commonly used protein thanks to its stable and species-independent production, and therefore can be highly useful for this application (Zhang et al., 1996). Specifically, its excitation and emission wavelengths are 500 and 513 nm, respectively (Lambert, n.d.).

However, similarly to what happens with absorbance, albeit it is a helpful way to visualize protein production for cells, it is difficult to quantify in an exact way the amount of produced protein *per cell*. This is because the result, presented in RFUs, only gives an idea of the protein quantity in the entire medium, but not how much each individual cell is producing.

Therefore, an alternative metric derived from fluorescence that can be used is the Molecules of equivalent fluorescein (MEFL).

Molecules of equivalent fluorescein (MEFL)

The equivalence from fluorescence to MEFL can be obtained by applying Equation 3.3 (Boada et al., 2019). In this case, the metric used to normalize these units is not directly the fluorescence, but the Normalized Fluorescent Radiance (NFR), which is derived from fluorescence.

$$\begin{aligned} &\text{MEFL} = q_0 \cdot (\text{NFR})^{q_1} \\ &\text{where} \quad q_0 = 10^{12.28} \quad \text{and} \quad q_1 = 1.038 \end{aligned} \tag{3.3}$$

As for the particles, a linear relationship can be established between the NFR and the number of fluorescent molecules (Figure 3.4).

3.5.3 Molecules of equivalent fluorescein per particle

Now, a performance metric that derives from both of the previously mentioned (particles and MEFL) is the MEFL/particle.

Since the MEFL allows to estimate the amount of fluorescein molecules in a certain culture, it means that it only gives an idea of how much fluorescent protein is produced globally – however, it might be interesting to quantify how many protein each cell is producing individually. For this reason, this metric can be obtained as a fraction between MEFL and particles (Equation 3.4).

$$\text{MEFL per particle} = \frac{q_0}{p_0} \cdot \frac{(\text{NFR})^{q_1}}{(\text{Abs}_{600})^{p_1}} \tag{3.4}$$

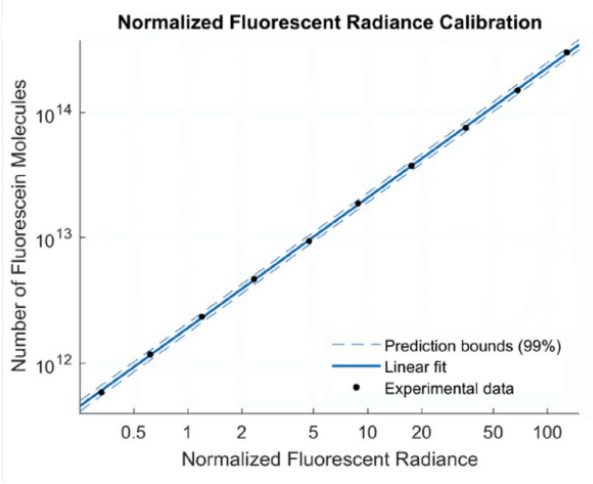


Figure 3.4: Equivalence between NFR and the amount of fluorescein molecules in the studied medium. Graph taken from Boada et al., 2019.

3.6 Identification of systems

Broadly speaking, the characterization of genetic circuits consists in the adjustment of parameters that define mathematical models using experimental data. Ideally, the model is then used to predict the behavior of the biological system for which it was created. This allows for the process of constructing genetic circuits to be based on informed decision-making through computational simulations instead of trial and error.

However, this is not an easy task, primarily due to the fact that it involves working with living cells whose behavior is not easily predictable, and the reproducibility of these systems is not as straightforward as, for example, that of electrical systems.

On the one hand, it is important to consider that introducing a synthetic circuit that triggers the production of a non-native protein into a cell forces said organism to utilize its resources (especially for transcription and translation processes) for purposes that are not inherent to it. This can impose a significant metabolic burden that needs to be taken into account when mathematically modeling the system.

Nevertheless, this can be incorporated into mathematical models in a relatively easy way by considering the cell's growth rate, which can be obtained from experimental data and calculated by means of the metrics described in Section 3.5.

On the other hand, it is crucial to take into account the context of the system and how the model parameters are being obtained. While the most common approach is the isolated characterization of specific BioBricks through their steady-state responses, this approach is not effective for models where their behavior is integrated into more complex circuits, as it is necessary to consider their interaction with the rest of the components (Boada et al., 2019).

It is for this reason that in this project it was decided to characterize the basic parts described in Chapter 2 in a global manner. In other words, the process of obtaining the final model will focus on characterizing all the parameters simultaneously, instead of attempting to find every parameter by adapting it to each specific part. Ultimately, the aim is to ensure that the resulting

model is consistent with the experimental data and cohesive with the rest of the species in the system.

Thus, the system identification requires an optimization process, as shown in Figure 3.5, which can be summarized in three fundamental steps:

1. **Problem definition.** The context is defined and a cost function is selected.
2. **Optimization.** The cost function is minimized, which allows to refine the model's parameters.
3. **Decision making process.** Based on the results, the mathematical model is completed.

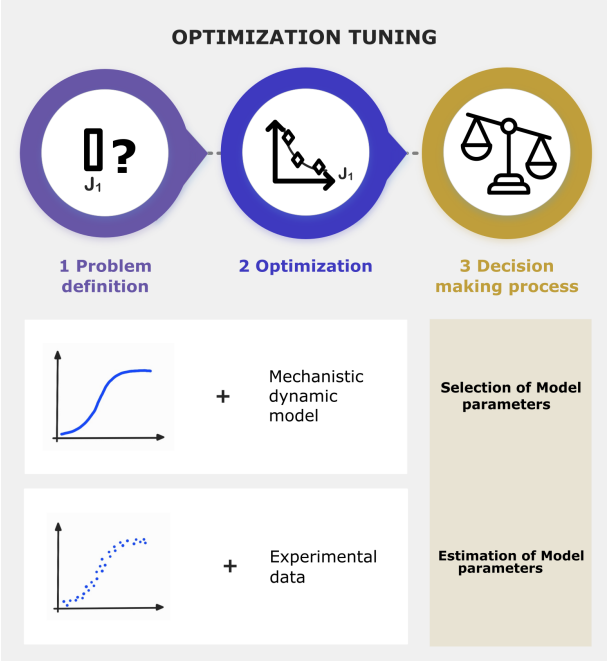


Figure 3.5: Schematic representation of the system identification process.

This sets the grounds for the characterization process, and allows for the successful obtainment of a digital twin that will accurately describe the real-world system's behaviour.

Chapter 4

Materials and Methods

4.1 Workflow: the DBTL cycle

The workflow that will be followed during this project will be the **Design, Build, Test, and Learn (DBTL) cycle**, which aims to streamline and optimize the design-build-test-learn of gene circuits in synthetic biology.

The iterative nature of this cycle allows for continuous refinement and improvement, as insights gained from experimental data inform the subsequent design iterations. This iterative approach not only promotes better understanding of the system's behavior but also facilitates the identification and fine-tuning of the parameters necessary for accurate modeling and practical implementation in synthetic biology.

The cycle involves four key steps, as shown in Figure 4.1.

- **Design**. In this first step, the aims of the project are established. The genetic constructs and the resulting synthetic gene circutis are computationally designed. Also, a mathematical model is developed in order to predict its behaviour and anticipate its response to different stimuli.
- **Build**. In the building phase, the designed synthetic gene circuits are assembled in the laboratory, and introduced into host organisms through cellular transformation (see Section 4.4.2).
- **Test**. Once the gene circuit is assembled, it has to be tested in the laboratory through the conduction of experiments to probe if it is properly working.
- **Learn**. The results of the Test phase are analyzed and compared with the results from the Design phase. The insights learned from the in-vivo experiments can help us improve the design of the initial version of the gene circuit. In this context, we can re-configurate the design, changing some of the biological parts, or tune the parameters of the mathematical model. Then, the DBTL cycle starts again.

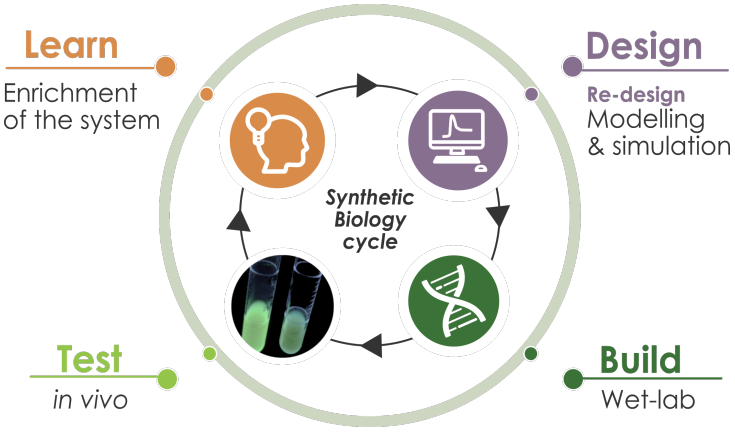


Figure 4.1: Depiction of the Design, Build, Test, and Learn cycle steps. (Taken from "Ecuador iGEM Team", 2021).

4.2 Mathematical approach

As proposed by Rolf Johansson in 1993, modeling and parameter identification of a system can be decomposed into a series of processes. This also can be associated with the Design, Build, Test, and Learn cycle, as shown in Figure 4.2.

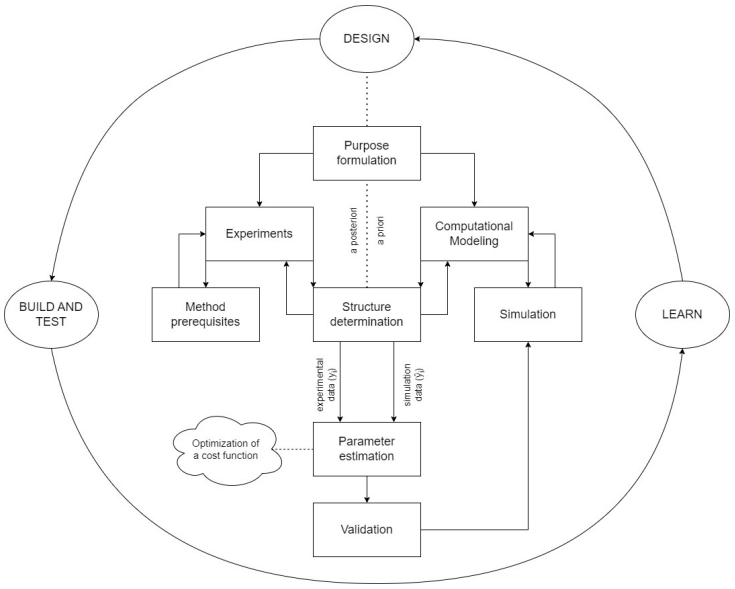


Figure 4.2: Steps of a system identification process. Diagram inspired from Johansson, 1993.

This way, we work with three central steps: structure determination, parameter estimation, and validation.

1. **Structure determination** refers to the fact that, once the model is generated and the experiments are conducted, it is important to establish the algorithms and software that will be used to perform the system identification – meaning, the environment in which this will take place and the techniques that will be employed for it.

2. **Parameter estimation** is the actual system identification process, in which some of the parameters that are unknown can be computed.
3. **Validation** allows to compare the mathematical model with experimental behaviour to assess the success of the model.

4.2.1 Law of Mass Action (LAM)

In this subsection, several key aspects of the mathematical approach of this project will be discussed: the principles for generating the model equations and how, ultimately, the presented results have been achieved.

The law of mass action states that in a chemical reaction, the rate at which this reaction occurs can be described as the product of the concentrations of the reactants, taking into account their stoichiometry (Érdi and Tóth, 1989). While the LAM is generally used to define the overall behavior of a reaction, it can also be applied to determine the temporal variation of the concentration of a specific species involved in the process (Picó Marco, 2021).

For this project, this will be the approach taken since the interest lies in understanding and modeling the behaviour of specific species to later combine these equations.

Reaction 4.1 shows a chemical reaction in which the biochemical species n_1 and n_2 are transformed into n_3 at a rate of r_1. By applying the LAM, Equation 4.2 is obtained.

$$\mathrm{n}_1 + \mathrm{n}_2 \underset{r_1}{\overset{r_2}{\leftrightarrows}} \mathrm{n}_3 \tag{4.1}$$

$$\dot{n_3} = r_1 \cdot n_1 \cdot n_2 - r_2 \cdot n_3 \tag{4.2}$$

In a general case, this can be expressed mathematically as shown in the set of Equations 4.3.

$$\begin{aligned} &aA + bB \underset{r_x}{\overset{r_y}{\leftrightarrows}} cC \\ &\frac{d[A]}{dt} = -r_x \cdot [A]^a \cdot [B]^b + r_y \cdot [C]^c \end{aligned} \tag{4.3}$$

Therefore, for any chemical reaction, it is possible to describe the behaviour of the species involved by applying the LAM.

4.2.2 Hill equation

The Hill equation, proposed by Archibald Hill in 1910, is a widely used function in the field of biochemistry to describe the interaction between a ligand and its receptor. It provides insight into the affinity between these biochemical species. Similarly, a ligand can be a species known as a Transcription Factor (TF), and a receptor corresponds to an inducible promoter that starts the transcription process.

Reaction 4.4 represents the binding-unbinding process among the TF, a protein (M) and an inducer (Ind). As observed, it is taken into account that these molecules can exist in various forms (monomer, dimer, etc.) – meaning, they may not necessarily be present individually.

$$\mathrm{n}_m \cdot \mathrm{M} + \mathrm{Ind} \underset{k_\alpha}{\overset{k_\beta}{\longleftrightarrow}} \mathrm{TF} \tag{4.4}$$

By applying the Law of Mass Action, Equation 4.5 is obtained, which represents the temporal variation of the concentration of the TF.

$$\dot{TF} = k_\alpha \cdot M^{n_m} \cdot Ind - k_\beta \cdot TF \tag{4.5}$$

It is worth noting that this process occurs so rapidly that it can be assumed that the concentration of the TF is constant over time, and therefore its temporal variation is practically negligible (Fraile López, 2020). Reducing the derivative to zero is commonly referred to as a Quasi-Steady-State Approximation. Thus, by setting TF = 0 and isolating TF, Equation 4.6 is obtained. It is important to highlight the substitution of the fraction between the coefficients k_α and k_β with the constant k_{dis}, known as the dissociation constant of the TF complex.

$$\begin{aligned} TF &= \frac{M^{n_m} \cdot Ind}{k_{dis}} \\ \text{where} \quad k_{dis} &= \frac{k_\beta}{k_\alpha} \end{aligned} \tag{4.6}$$

Now, Equation 4.7 is the actual Hill equation (Hill, 1910), which represents the behaviour of the complex that results from the binding between the TF to the promoter (P.TF). In this case, the factor n is an estimator of the cooperativity between the TF and the promoter (Goutelle et al., 2008).

$$P.TF = \frac{TF^n}{(k_d)^n + TF^n} \tag{4.7}$$

It is important to highlight the difference between constants k_{dis} and k_d. While k_{dis} represents the affinity between the elements that form the TF (roughly, the rate at which they associate and dissociate), k_d provides the concentration of inducer molecules needed to achieve half of the binding sites being occupied (Gesztelyi et al., 2012).

Another aspect to highlight is that the number of binding sites relies on the number of plasmids (or amount of DNA) present in the cell, which is determined by the Copy Number (C_N) as described in previous sections. Therefore, this should also be reflected in the equation, resulting in Expression 4.8 (Trabelsi et al., 2018).

$$P.TF = \frac{TF^n}{(k_d \cdot C_N)^n + TF^n} \tag{4.8}$$

Finally, by substituting Equation 4.5 into the P.TF expression, the modified Hill equation is obtained (see Equation 4.9), to accurately model an inducible promoter.

$$P.TF = \frac{Ind^n}{(\frac{k_{dis} \cdot k_d \cdot C_N}{M^{n_m}})^n + Ind^n} \tag{4.9}$$

4.2.3 Optimization

A common approach for parameter identification and characterization in mathematical models is the use of optimization methods, where the objective is usually minimizing a cost function. This cost function represents the error between the predictions made by the mathematical model and experimental data.

It is crucial to emphasize the importance of selecting an appropriate cost function, as there can be different error metrics depending on the nature of the experimental data. It is essential for it to closely fit the data in order to effectively minimize the error.

Therefore, it is important to carefully analyze the data before optimization. For example, if it is known that the set is significantly noisy, it may be necessary to prioritize a function that is robust to outliers.

In this project, we chose the Root Mean Logarithmic Squared Error (RMLSE) as an error metric for three main reasons (M, 2021):

- **Robustness to outliers.** When collecting data from cell cultures, there may be anomalous data points due to cross-contamination between cultures or values that fall outside the range for certain metrics due to equipment limitations.
- **Scale independence.** Errors are normalized, so an error of 100 units between 1000 and 900 is interpreted in the same way as an error of 10 units between 100 and 90. This is important in this case because fluorescence measurements are initially very low but grow dramatically after induction. Thus, it is necessary to consider this fact to avoid dismissing these cases incorrectly.
- **Greater penalty for underestimation.** While in the literature this may be presented as a disadvantage, in this case it is considered an advantage. Protein production is a process where the transition between a low production level and a higher one occurs suddenly. Thus, it is necessary to penalize underestimation so that this 'jump' in production is not overlooked.

Thus, for the calculus of the cost function, Expression 4.10 was used, where y_i represents the true value and $\hat{y}_i$ the predicted one.

$$\text{RMLSE} = \sqrt{\sum_{i=1}^{N} (\log_{10}(y_i + 1) - \log_{10}(\hat{y}_i + 1))^2} \tag{4.10}$$

On the other hand, it is relevant to discuss the optimization method, which is selected using the same criteria as the cost function. In this case, the genetic algorithm (GA) will be used, which is inspired in the concept of Darwinian evolution.

Roughly, the steps followed during this optimization are:

1. **Population Initialization.** It begins with an initial population of individuals, where each individual represents a possible solution to the system to be solved.

2. **Fitness Value Assignment.** A fitness value is calculated to evaluate the accuracy of the solution for each individual and is then transformed into *expectation values*.
3. **"Parent" Selection.** Individuals with higher expectation values are selected to produce new individuals.
4. **"Children" Production from Parents.** Selected individuals are combined through *crossover* or altered through *mutation* with a certain probability p. This results in new individuals with solutions that slightly differ from the ones obtained in the first step.
5. **Repetition of Steps.** Steps 2, 3, and 4 are iteratively repeated until a *termination criterion* is met.
6. **Termination Criterion.** This criterion can be a maximum number of iterations (or generations), a time limit for optimization, the reaching of an error below a certain tolerance, among others.

These steps together constitute the genetic algorithm, which aims to find an optimal solution by iteratively evolving a population of individuals through selection, reproduction, and variation of solutions ("How the Genetic Algorithm Works - MATLAB & Simulink", n.d.; Yang, 2021).

4.3 Computational environment

The computational environment played a critical role in the development of this project, as every phase of the Design, Build, Test, and Learn cycle was conducted by a computational approach.

4.3.1 Hardware

To conduct the simulations, a computer with the characteristics specified in Table 4.1 was used.

Component	Characteristics
Processor	AMD Ryzen™ 7
RAM	16.00 GB
Operative System	Windows 11 Home
Hard drive storage	500 GB

Table 4.1: Description of the components of the computer used to conduct the simulations.

4.3.2 Software

Several tools were employed for different purposes. Below, a brief description will be provided for these computational programs and their specific function in this project.

MATLAB

Version R2022a of the MATLAB programming environment was used for conducting of simulations and code editing of the mathematical model of each gene circuit.

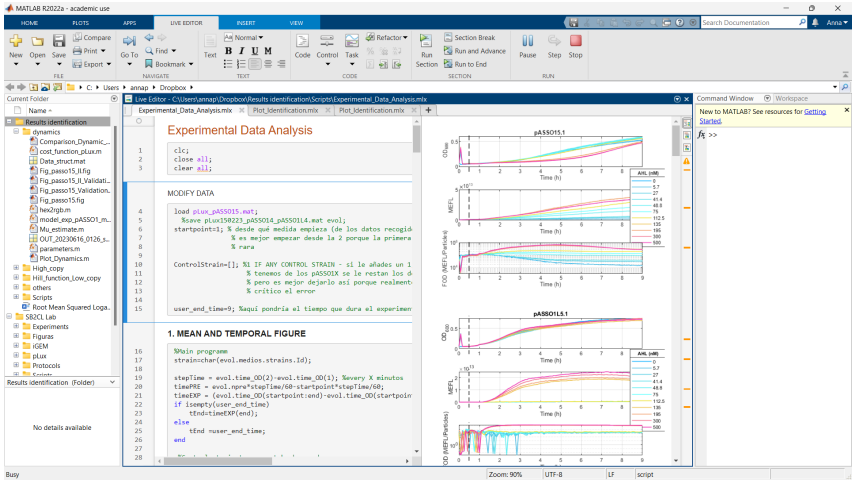


Figure 4.3: Screenshot of the MATLAB environment.

Benchling

Benchling is a an online platform designed primarily for use in Research and Development laboratories, providing a wide range of useful functions for experiment planning and development of the DNA assembly ("Cloud-based platform for biotech R&D | Benchling", n.d.). Specifically, it was used for the following purposes:

- Experiment planning and management
- Gene circuits and plasmids construction
- Primer design for PCRs
- Recording experimental result data

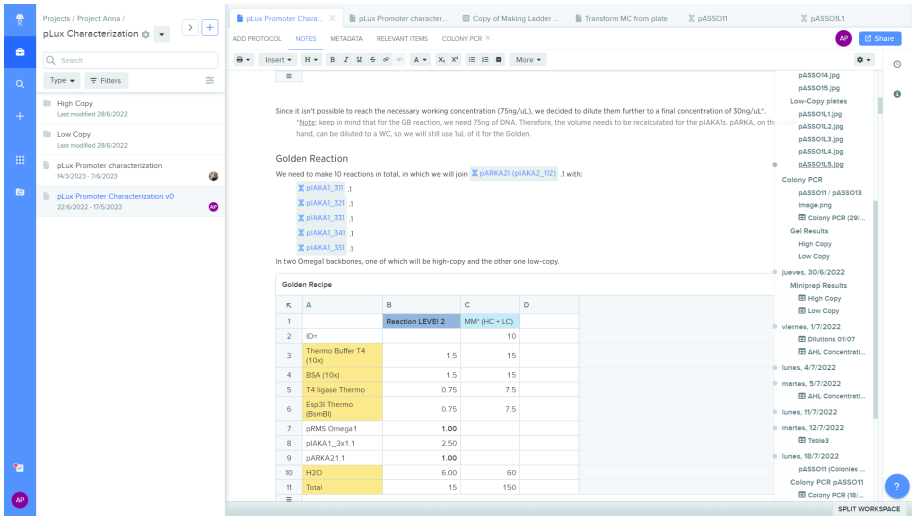


Figure 4.4: Screenshot of the Benchling environment.

BioTek Gen5

BioTek Gen5 is the software of the Cytation instrument described in Section 4.4.1. It allows data processing and preliminary visualization of the experimental readings obtained from the Cytation (such as OD and fluorescence).

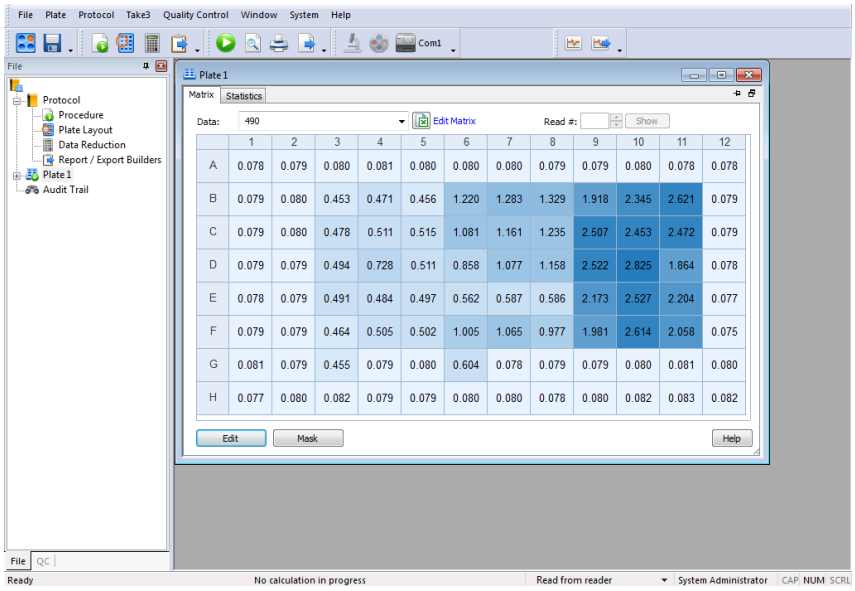


Figure 4.5: Screenshot of the BioTek Gen5 environment.

4.4 Experimental environment

For the development of the project, several pieces of equipment to build the gene circuits, manipulation of the cell cultures containing the gene circuits, and monitoring their behaviour during the conducted experiments were used.

4.4.1 Equipment

The equipment listed was used to conduct the experimental protocols that will be explained in the following subsection.

Thermocycler

The model employed is the TECHNÉ 3Prime Thermal Cycler by Jeulin ("TECHNÉ 3Prime Thermal Cycler", n.d.). It was mainly used to conduct the Golden Braid (GB) Assembly protocol.

Figure 4.6: TECHNÉ 3Prime Thermal Cycler (thermocycler).

Opentrons OT-2 Robot

Developed by Opentrons Labwork Inc., it is a robot that enables automation of pipetting processes in the laboratory ("OT-2 Liquid Handler | Opentrons Lab Automation | Opentrons", n.d.). In the context of this project, specific protocols designed in the lab were used to perform the initial dilutions required for cell culture preparation, as well as to carry out inductions with a proper inducer during the appropriate phase of the experiment.

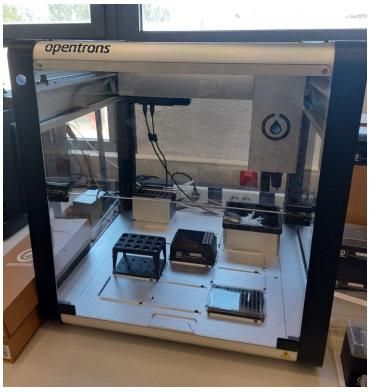


Figure 4.7: Opentrons OT-2 by Opentrons Labwork Inc.

BioTek Cytation™3 Imaging Reader

Developed by Agilent Technologies, this device was primarily used for its integrated spectrophotometer to perform readings of optical density (OD) as well as green fluorescence of culture plates during experiments.

Figure 4.8: Cytation 3 Imaging Reader by Agilent Technologies.

Orbital shaker incubator

The used model is the Stuart SI500. It was used for the growth of *E. coli* cell cultures in liquid medium. The set temperature for it was of 37ºC, at 230 revolutions per minute (rpm).

Figure 4.9: Stuart SI500 orbital shaker incubator.

Incubator

The used model is the Memmert IN110 incubator. It incubator was used for the growth of solid cell cultures of *E. coli* in Petri dishes. The set temperature for it was of 37ºC.

Figure 4.10: Memmert IN110 incubator.

4.4.2 Protocols

The experimental protocols employed in this project played a crucial role in the precise manipulation and assembly of genetic elements, as well as their integration into living cells. Moreover, additional techniques were used for the analysis and verification of the genetic circuits, ensuring the validity of the constructed designs.

Golden Braid Assembly

Aim: bring together fragments of DNA, resulting in Level 0, Level 1, or Level 2 assemblies, depending on the components and restriction enzymes employed (Sarrion-Perdigones et al., 2011).

The necessary components to conduct the GB Assembly are listed in Table 4.2. Notice how two different types of endonucleases can be used – when constructing assemblies where the destination vector is of type α, the used enzyme will be BsaI; however, for destination vectors of type Ω, BsmBI will be used.

Another key point is that the DNA component refers to every part to use. For example, a Level 2 assembly is built using two Level 1 assemblies and an Ω destination vector – therefore, 75 ng of each component are needed. Since these are suspended in water, it is necessary to know its concentration, and add the according amount (which also explains why the amount of water is not fixed).

Now, all elements must be combined in a single PCR tube (one per construction). When the tubes are ready, put them in the Thermocycler and run a program with the steps described in Table 4.3. After finishing, the plasmids will be ready to be incorporated into *E. coli* bacteria by means of cellular transformation (see Protocol Transformation of chemo-competent *Escherichia coli* cells).

Transformation of chemo-competent *Escherichia coli* cells

Aim: incorporate the created plasmid into *E. coli* cells.

The necessary materials are:

Component	**Amount** (µL)
T4 Ligase Buffer (10X)	1.50
BSA Buffer (10X)	1.50
T4 Ligase	0.75
BsaI or BsmBI Endonuclease	0.75
DNA (2 or more)	75 ng of each part used
H_2O	up to 15 µL
Total Volume	15

Table 4.2: Components for a GB assembly of genetic circuits.

Temperature (ºC)	**Time (min)**	**Number of cycles**
37	10	1
37	3	30
16	4	
50	10	1
80	10	1

Table 4.3: Thermocycler steps for the Golden Braid Assembly.

- Chemically competent cells
- Room temperature SOC medium
- Falcon tubes
- Tubes cooler
- Incubator
- Thermocycler

Start by setting up the necessary program on the Thermocycler. Turn off the lid pre-heating option and set the following steps:

1. Heat-shock: set the temperature at 50ºC for 25 seconds.
2. Post-incubation: set the temperature at 4ºC for 5 minutes.

Now, add between 2 and 3 uL of the assembled DNA directly into the tubes with the chemo-competent cells, and mix by pipetting up and down. Incubate them at 4ºC for 30 minutes in the fridge.

Once the time passes, put the tubes in the Thermocycler and run the program. When finished, add 450 uL of the SOC medium to each tube, mix by pipetting up and down, and transfer to a Falcon tube. Incubate at 37ºC for 1 hour (plasmids in high-copy vectors) or 2 hours (plasmids in low-copy vectors).

Preparation of a cell culture

Aim: grow cellular colonies derived from an overnight cell culture suspended in liquid medium in a Petri dish for further study and analysis.

The necessary materials are:

- Overnight cell culture suspended in liquid medium
- Petri dish with solid LB medium and antibiotic
- Agar beads
- Incubator

Start by choosing a Petri dish with solid LB medium and the adequate antibiotic based on the plasmid that the cell culture contains. For cells with α backbones, this will be kanamycin, whereas for Ω backbones it will be spectinomycin. This selection allows to make sure that only those cells that have incorporated the plasmid during cellular transformation will grow.

Add the desired amount of cell culture to the plate. For this project, 100 µL were added to each one. Now, throw a couple of agar beads into the plate and cover it. Shake gently until the culture is distributed across the entire plate.

Remove the beads and put the plate in the incubator at 37^{o}C overnight (approximately 16 hours). Note that if the used backbones are α or Ω, the colonies derived from cells that did not incorporate a plasmid with the desired Transcription Unit will appear as red.

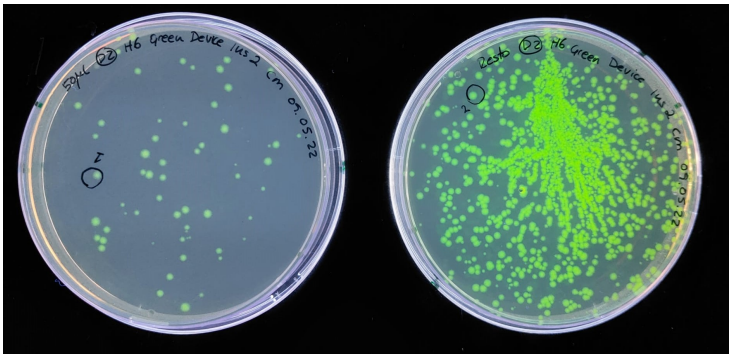


Figure 4.11: Example of cell cultures observed in an UV transilluminator.

Agarose gel electrophoresis

Aim: validation of DNA sequences according to size.

The necessary materials are:

- Agarose
- SB Buffer
- Purple Loading Dye (6X) by New England BioLabs
- Thermo Scientific™1 kb DNA GeneRuler
- DNA samples

- Erlenmeyer flask
- Electrophoresis apparatus
- Electrophoresis combs
- Microwave
- UV Transilluminator

To prepare 80 mL of gel with a 1% porosity, start by adding 0.8 g of agarose and 80 mL of SB buffer to an Erlenmeyer flask. Now, put the flask in the microwave at medium power and heat the mix until the solution becomes clear. Keep in mind that during heating the mix may come to a boil, so stopping the microwave periodically in order to avoid the liquid from evaporating is recommended.

Once the solution is clear, let it cool down to 50ºC. Put the necessary combs in the gel tray and add the entirety of the flask to it. Wait until the texture goes from liquid to gel-like and remove the combs. This will leave small wells in the gel (the amount depends on the comb size). Add SB buffer to the electrophoresis tank until the gel wells are no longer visible from above.

Load 6 µL of the GeneRuler into the first and (optionally) last well. Then, mix 8 µL of DNA sample with 2 µL of the Purple Loading Dye and load it into the following well. Repeat this second step until all samples are loaded in the gel.

Once loaded, connect the electrophoresis apparatus to the current and run at 230 V for 30 minutes.

When finished, grab the gel using latex gloves and put it on the transilluminator. Turn on the UV light and evaluate the results taking into account the desired size of the insert by comparing it to the ladder equivalences as shown in Figure 4.12.

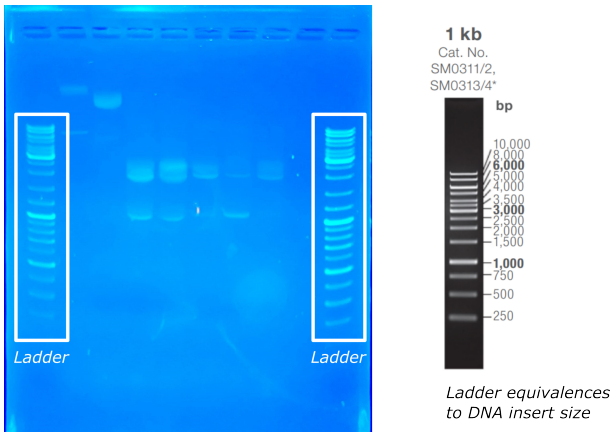


Figure 4.12: Example of the result of an electrophoresis gel and the reference ladder.

Colony PCR (cPCR)

Aim: amplify a region of interest delimited by primers in a DNA sequence to check the right size of a genetic circuit.

Primers are short DNA sequences designed to define the region of interest that will be amplified during the PCR process.

The necessary components and their amounts for a single colony reaction are listed in Table 4.4. Additionally, a thermocycler and PCR tubes are needed.

Component	**Amount (µL)**
10X Standard Taq Buffer	2
dNTPs (10 mM)	2
Forward Primer (10 µM)	0.4
Reverse Primer (10 µM)	0.4
Taq DNA Polymerase	0.1
Colony inoculated in water	5
ddH_2O	10.1
Total Volume	20

Table 4.4: Necessary components for the cPCR (for one colony).

Once all the components are united in PCR tubes (one per each colony to analyze), put them in the Thermocycler and run a program with the steps listed in Table 4.5. When finished, use an agarose gel to analyze results (see Protocol Agarose gel electrophoresis).

Step	**Temperature (^{o}C)**	**Time**	**Number of cycles**
Initial denaturation	95	2 min	1
Denaturation	95	20 sec	30
Annealing	56	1 min	
Extension	68	1 min per kb	
Final extension	68	5 min	1

Table 4.5: cPCR Thermocycler program steps.

Glycerol stock preparation

Aim: prepare a bacterial colony to be stored in a -80^{o}C freezer for future uses.

The materials needed are:

- 2 mL cryovial
- Glycerol 80%
- Overnight cell culture
- Vortex mixer

Start by adding 1440 µL of the overnight cell culture to the cryovial. Then, add 360 µL of the 80% glycerol. Vortex the tube and store in the freezer at -80^{o}C.

Part III

RESULTS

Chapter 5

Phase I: Design

5.1 Mathematical model

The aim is to design a digital twin or mathematical twin, which will be a set of ODEs that depict the behaviour of the gene circuit of interest.

In this case, the circuit of interest produces two proteins: LuxR (constitutively) and the Green fluorescent protein (GFP) (regulated by a Transcription factor). Figure 5.1 shows this process, which has two steps:

1. **Production of LuxR**

   The first step is the constitutive production of the LuxR protein, under the promoter J23106. The amount of LuxR protein depends on the combination of the promoter and RBS.

2. **AHL induction**

   When the N-Acyl homoserine lactone (AHL) molecule is induced in the liquid medium, it will be able to get through the membrane of the cell to interact with LuxR. The binding between these molecules (AHL and LuxR) forms the LuxR·AHL Transcription factor. Once this TF joins the Plux promoter, the transcription process of GFP will begin.

   Therefore, in this case the amount of produced protein will be regulated by the strength of the promoter Plux and the RBS, as well as by the AHL concentration inside the cell. Furthermore, the RBS will vary depending on the gene circuit, this is, five different levels of GFP produced since there are five different RBSs.

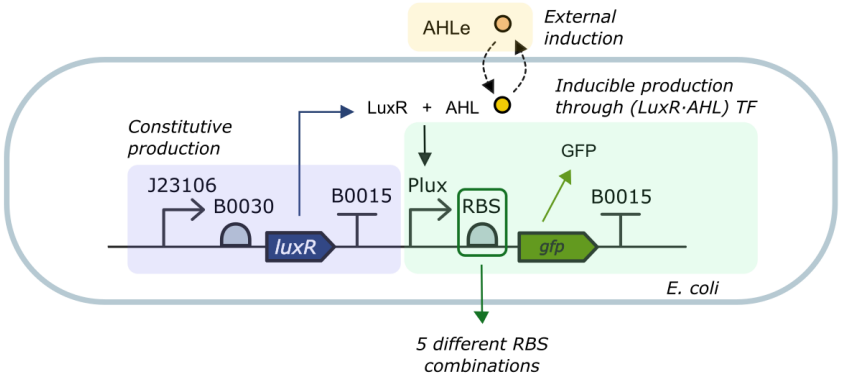


Figure 5.1: Behaviour of the genetic circuit built in the lab. LuxR is produced constitutively and GFP depends on the TF LuxR·AHL. Note how there are five different circuits depending on the RBS used.

5.1.1 Constitutive production

The set of biochemical Reactions 5.1 to 5.5 represent the constitutive synthesis of LuxR. As seen in Reaction 5.1, the promoter J23106 (pJ), allows the transcription of the DNA sequence into Messenger ribonucleic acid (mLuxR). The transcription rate is k_1 (min^{-1}), and the mLuxR is then translated into the protein LuxR by the ribosomes at the translation rate k_2 (min^{-1}).

However, it is also important to consider the degradation and dilution of both the mRNA and protein (Reactions 5.2 to 5.5). The degradation of the biological happens naturally at a rate of d_1 (min^{-1}) for mLuxR, and d_2 (min^{-1}) for LuxR. On the other hand, dilution is caused in the cell by cell division (cell growth) at a rate of μ (min^{-1}).

$$\text{pJ} \xrightarrow{\text{k}_1} \text{mLuxR} \xrightarrow{\text{k}_2} \text{LuxR} + \text{mLuxR} \tag{5.1}$$

$$\text{mLuxR} \xrightarrow{\text{d}_1} \emptyset \tag{5.2}$$

$$\text{mLuxR} \xrightarrow{\mu} \emptyset \tag{5.3}$$

$$\text{LuxR} \xrightarrow{\text{d}_2} \emptyset \tag{5.4}$$

$$\text{LuxR} \xrightarrow{\mu} \emptyset \tag{5.5}$$

Now, by applying the Law of Mass Action (LAM), the Model 5.6 is obtained. It represents the biochemical reactions in terms of ODEs. Note that this is a representation of the variation in time of the concentration of the three main biological species that participate in the constitutive expression of LuxR.

$$\begin{aligned} [\dot{pJ}] &= k_1 \cdot [pJ] \\ [\dot{mLuxR}] &= k_1 \cdot [pJ] - (d_1 + \mu) \cdot [mLuxR] \\ [\dot{LuxR}] &= k_2 \cdot [mLuxR] - (d_2 + \mu) \cdot [LuxR] \end{aligned} \tag{5.6}$$

Now, notice how the variation of the promoter concentration depends only on the concentration of the promoter itself, and is regulated by its transcription rate k_1.

On the other hand, the variation of the mRNA coding for LuxR also depends on the concentration of the DNA pJ, as well as on the degradation (d_1) and dilution (μ) rates of the mRNA itself. The same happens with protein LuxR, although in this case it is dependant on the rate at which it is translated (k_2), and on its degradation (d_2) and dilution (μ) rates.

5.1.2 Inducible production

Unlike LuxR, the transcription of the Green fluorescent protein (GFP) is regulated by an inducible promoter, and requires the presence of a TF to activate it. Therefore, the reactions that represent the GFP's production will be slightly different than the ones above, since it is necessary to consider additional reactions regarding the TF.

$$2\,\mathrm{LuxR} + 2\,\mathrm{AHL} \underset{k_\beta}{\overset{k_\alpha}{\leftrightarrows}} (\mathrm{RA})_2 \tag{5.7}$$

$$\mathrm{Plux} + (\mathrm{RA})_2 \underset{k_{off}}{\overset{k_{on}}{\leftrightarrows}} (\mathrm{Plux} \cdot (\mathrm{RA})_2) \tag{5.8}$$

$$(\mathrm{Plux} \cdot (\mathrm{RA})_2) \xrightarrow{\mathrm{k_3}} \mathrm{mGFP} + (\mathrm{Plux} \cdot (\mathrm{RA})_2) \tag{5.9}$$

$$\mathrm{mGFP} \xrightarrow{\mathrm{k_4}} \mathrm{GFP} + \mathrm{mGFP} \tag{5.10}$$

The initial step is the Reaction 5.7, that involves the binding of two LuxR proteins with two molecules of AHL. This interaction leads to the formation of the transcription factor (TF), denoted as $(RA)_2$. It is worth noting that this process is reversible, because these species are continuously binding and unbinding at rates k_α (min^{-1}) and k_β (min^{-1}) respectively.

Reaction 5.8 denotes the interaction between the promoter Plux and the TF, in which the binding and unbinding of both parts occurs at constant rates k_{on} (min^{-1}) and k_{off} (min^{-1}).

In regards to transcription (Reaction 5.9) and translation (Reaction 5.10), it is considered that they happen at rates k_3 (min^{-1}) and k_4 (min^{-1}), and both processes are irreversible as in the case of constitutive production.

By applying the LAM, Model 5.11 is obtained.

$$\begin{aligned}
[\dot{Plux}] &= -k_{on} \cdot [Plux] \cdot [(RA)_2] + k_{off} \cdot [(Plux \cdot (RA)_2)] \\
[\dot{(RA)}_2] &= -k_{on} \cdot [Plux] \cdot [(RA)_2 + k_{off}] \cdot [(Plux \cdot (RA)_2)] \\
[(Plux \cdot \dot{(RA)}_2)] &= k_{on} \cdot [Plux] \cdot [(RA)_2] - k_{off} \cdot [(Plux \cdot (RA)_2)] \\
[(Plux \cdot \dot{(RA)}_2)] &= -k_3 \cdot [(Plux \cdot (RA)_2)] \\
[\dot{mGFP}] &= k_3 \cdot [(Plux \cdot (RA)_2)] - (d_3 + \mu) \cdot [mGFP] \\
[\dot{GFP}] &= k_4 \cdot [mGFP] - (d_4 + \mu) \cdot [GFP]
\end{aligned} \tag{5.11}$$

5.1.3 Complete model

Considering Sections 5.1.1 and 5.1.2, the complete model of the gene circuit can be defined. However, in order to simplify the expressions, the species names were simplified as shown in Table 5.1.

Symbol	Species	Description
x_1	pJ	Constitutive promoter J23106
x_2	mLuxR	LuxR protein mRNA
x_3	LuxR	LuxR protein
x_4	Plux	Inducible promoter Plux
x_5	$(RA)_2$	Transcription factor (TF)
x_6	$(Plux{\cdot}(RA)_2)$	Binding between Plux and TF
x_7	mGFP	GFP protein mRNA
x_8	GFP	GFP protein

Table 5.1: Species equivalence.

The result is Model 5.12. However, the problem now is that, albeit accurate, this ODE model is overparameterized, and many of these parameters are unknown. Furthermore, there are several species which are difficult to measure in an experimental environment (e.g., the amount of any form of mRNA or the binding between promoter and TF).

For this reason, a **model reduction** is needed in order to simplify the equations.

$$\begin{aligned}
\dot{x_1} &= k_1 \cdot x_1 \\
\dot{x_2} &= k_1 \cdot x_1 - (d_1 + \mu) \cdot x_2 \\
\dot{x_3} &= k_2 \cdot x_2 - (d_2 + \mu) \cdot x_3 \\
\dot{x_4} &= -k_{on} \cdot x_4 \cdot x_5 + k_{off} \cdot x_6 \\
\dot{x_5} &= -k_{on} \cdot x_4 \cdot x_5 + k_{off} \cdot x_6 \\
\dot{x_6} &= k_{on} \cdot x_4 \cdot x_5 - k_{off} \cdot x_6 \\
\dot{x_6} &= -k_3 \cdot x_6 \\
\dot{x_7} &= k_3 \cdot x_6 - (d_3 + \mu) \cdot x_7 \\
\dot{x_8} &= k_4 \cdot x_7 - (d_4 + \mu) \cdot x_8
\end{aligned} \tag{5.12}$$

5.1.4 Model reduction

When taking into account all reactions involved, the resulting models obtained are often over-parametrized, thus increasing the number of states in which the system can be. This makes the identification of its parameters more difficult.

For this reason, model reduction supposes an important aspect when developing a mathematical model. Thus, there are two main assumptions that may allow to reduce a model:

1. **System invariants**

System invariants exist because, in certain biological situations, it can be assumed that the concentration of a particular species remains constant (therefore, $\dot{x} \simeq 0$). This means that, when certain species have constant concentrations, it is implied that their production and degradation rates are balanced, resulting in no net change over time. By considering these species as constants, we can eliminate their time derivatives from the equations, thus simplifying the model.

2. **Quasi-Steady-State Approximation (QSSA)**

   In the quasi-steady-state approximation (QSSA), it is assumed that certain species in the system undergo rapid reactions and reach a steady-state concentration much faster compared to other species. This steady-state condition implies that their production and degradation rates are in equilibrium, resulting in no significant net change over time. This simplification allows us to focus on the dynamics of the slower-reacting species and analyze the system's behavior under this approximation (Segel and Slemrod, 1989).

The first system invariants to be noticed in the system are the constitutive promoter J23106 (x_1) and inducible promoter Plux (x_4).

As seen in Section 3, a backbone's Copy Number establishes the initial amount of DNA to be transcribed. Since the Transcription Unit of the backbone contains both promoters, their concentration can be considered equivalent to the C_N (Boada Acosta, 2018).

This way, it is possible to express these relations by the set of Equations 5.13. Notice how, in the inducible case, the promoter is considered in both its forms: free LuxR promoter and bound to the TF.

$$\begin{aligned} \dot{x_1} &\simeq 0 \\ x_1 &\simeq C_N \\ \dot{x_4} + \dot{x_6} &= 0 \\ x_4 + x_6 &\simeq C_N \end{aligned} \tag{5.13}$$

An additional assumption can be made regarding the Plux and TF binging $\dot{x_6}$. The process of the binding and unbinding of the promoter and its TF takes place in terms of seconds, in contraposition to transcription and translation, which can take up to minutes and hours, respectively. Therefore, the QSSA can be applied, assuming that $\dot{x_6} \simeq 0$.

This expression can be combined with the last equation of the previous set, and by isolating x_6, Equations 5.14 are obtained.

$$\begin{aligned} \dot{x_6} &= k_{on} \cdot (C_N - x_6) \cdot x_5 = 0 \\ x_6 &= \frac{k_{on} \cdot C_N \cdot x_5}{k_{on} \cdot x_5 + k_{off}} \end{aligned} \tag{5.14}$$

Moreover, the k_{on} and k_{off} parameters can be simplified by dividing both numerator and denominator by k_{on}, which results the **dissociation constant** k_d (Najafpour, 2007). The result is a Hill-like function (Equation 5.15), that relates the transcription process with the TF (x_5) that activates it.

$$\begin{aligned} x_6 &= C_N \cdot \frac{x_5}{k_d + x_5} \\ \text{where} \quad k_d &= \frac{k_{off}}{k_{on}} \end{aligned} \tag{5.15}$$

However, as described in the Equation 4.9 from Chapter 4, several additional considerations need to be taken into account to accurately model this interaction. This way, the inducer will be the AHL molecule, inside the cell, while M will be LuxR in this case. Now, the n and n_m exponents need to be defined as well.

For this system, the cooperativity index (n) is unknown, and therefore will be one of the parameters to estimate.

FOr n_m, Reaction 5.7 is considered. As seen in Chapter 4, this parameter can be interpreted as the amount of LuxR needed in relation to the amount of AHL inside the cell (AHL_{int}). Since both elements have the same stoichiometric coefficient, $n_m = 1$.

Thus, the new **Hill equation** for this system is Equation 5.16. Note that AHL is expressed in terms of concentration. Furthermore, it is worth noting that k_{dlux}, which could be considered as a parameter derived from k_d, is not equal to the dissociation constant. In this case, it represents the half-concentration, which is the concentration of inducer molecules required to ensure half of the binding sites are occupied to start the transcription process.

$$x_6 = C_N \cdot \frac{[AHL_{int}]^n}{(\frac{k_{dis} \cdot k_{dlux} \cdot C_N}{LuxR})^n + [AHL_{int}]^n} \tag{5.16}$$

Now, the QSSA can be applied to the equations representing transcription. Albeit occurring at a much slower pace in comparison to the binding and unbinding of the promoter and TF, it still is very fast when compared to the hours it takes for the translation process to be completed. For this reason, it can be considered that this process is also quasi-steady, and therefore, the variation of the concentration of both mLuxR and mGFP is assumed negligible.

This way, applying the QSSA to these species, Equations 5.17 are obtained.

$$\begin{aligned} \dot{x_2} &= k_1 \cdot C_N - (d_1 + \mu) \cdot x_2 = 0 \\ x_2 &= \frac{k_1 \cdot C_N}{d_1 + \mu} \\ \dot{x_7} &= k_3 \cdot C_N \cdot \frac{[AHL_{int}]^n}{(\frac{k_{dis} \cdot k_d \cdot C_N}{LuxR})^n + [AHL_{int}]^n} - (d_3 + \mu) \cdot x_7 = 0 \\ x_7 &= C_N \cdot \frac{k_3}{d_3 + \mu} \cdot \frac{[AHL_{int}]^n}{(\frac{k_{dis} \cdot k_{dlux} \cdot C_N}{LuxR})^n + [AHL_{int}]^n} \end{aligned} \tag{5.17}$$

Now, by substituting the expressions obtained above into the equations for the protein species $\dot{x_3}$ and $\dot{x_8}$ from Model 5.12, Model 5.18 is obtained. This is a reduced version of the complete model presented in Section 5.1.3. Note that an additional term was added for $\dot{x_8}$, which allows us to consider the basal production (α) of GFP when there is no TF.

$$\begin{aligned} \dot{x_3} &= C_N \cdot \frac{k_1 \cdot k_2}{d_1 + \mu} - (d_2 + \mu) \cdot x_3 \\ \dot{x_8} &= C_N \cdot \frac{k_3 \cdot k_4}{d_3 + \mu} \cdot (\alpha + (1 - \alpha) \cdot \frac{[AHL_{int}]^n}{(\frac{k_{dlux} \cdot k_i \cdot C_N}{x_3})^n + [AHL_{int}]^n}) - (d_4 + \mu) \cdot x_8 \end{aligned} \tag{5.18}$$

From Model 5.18, the measurable species are the AHL concentration, which is externally induced in each experiment; and the concentration of GFP (x_8), which is the measured MEFL/OD_{600} as expained in Section 3.5.2.

Now, it must be highlighted that the growth rate μ actually changes over time. Therefore, it is not actually a fixed parameter, but a time function $\mu(t)$. Thus, in Model 5.18 its value will be calculated from the Absorbance experimental data by means of the MATLAB function *Mu_ estimate* (see Appendix).

Regarding the parameters, it must be highlighted that some of them depend on the gene circuit. Specifically:

1. **Copy Number (C_N)**

   Since this is defined as the vector, two Copy Numbers must be estimated: One for the high-copy version, and One for the low-copy.

2. **GFP translation rate (k_4)**

   This rate is defined by the RBS, and therefore there will be five different values for this parameter.

3. **Half-concentration (k_{dlux})**

   The AHL concentration at which half of the binding sites are occupied is dependant on whether the circuit is high-copy or low-copy. This is due to the fact that said factor will determine the amount of DNA in the cell, and thus of the initial free promoter binding sites.

4. **mRNA for GFP degradation rate (d_3)**

   This parameter is also dependant on which version is analyzed: high-copy or low-copy.

5. **GFP protein degradation rate (d_4)**

   Since the degradation rate also depends on the amount of protein produced, it will also depend on whether the circuit is high-copy or low-copy.

6. **Cooperativity index (n)**

   This index, as mentioned previously, defines the affinity between the promoter and the TF. Thus, it will also depend on the amount of available promoters, and therefore again on the circuit being high-copy or low-copy.

5.1.5 AHL production

For this model, it is also important to consider the equations for the internal (inside the cell) and the external molecules of AHL (AHL_{int} and AHL_{ext}, respectively).

These species react according to Reactions 5.19 and 5.20, where D is the kinetic rate at which AHL is transported outside the cell, DV_c is the rate at which it diffuses inside the cell, and d_A and d_{Ae} is the degradation rate of the internal (AHL_{int}) and external (AHL_{ext}) amount of molecules.

$$\text{AHL}_{\text{int}} \underset{DV_c}{\overset{D}{\longleftrightarrow}} \text{AHL}_{\text{ext}} \tag{5.19}$$

$$\text{AHL}_{\text{int}} \xrightarrow{\text{d}_\text{A}} \emptyset \tag{5.20}$$

$$\text{AHL}_{\text{ext}} \xrightarrow{\text{d}_\text{Ae}} \emptyset \tag{5.21}$$

Thus, by applying the LAM to Reactions 5.19, 5.20, and 5.21, the set of Equations 5.22 is obtained to represent the dynamics of these species (Boada Acosta, 2018). Note that x_9 and x_{10} are AHL_{int} and AHL_{ext}, respectively.

$$\begin{aligned} \dot{x_9} &= D \cdot \frac{V_{cell}}{V_{ext}} \cdot x_{10} - (D + d_A + \mu) \cdot x_9 \\ \dot{x_{10}} &= -D \cdot n_{cell} \cdot \frac{V_{cell}}{V_{ext}} \cdot x_{10} + D \cdot \sum_{i=1}^{x_{11}} x_9 \end{aligned} \tag{5.22}$$

This introduces a series of new parameters, which are:

- *E. coli* cell volume (V_{cell})
- Culture medium volume (V_{ext})

The number of cells n_{cell}, on the other hand, is another equation which models how the cell population grows.

5.1.6 Cell population growth

Cell growth can be approximated through the logarithmic growth Model 5.23 (Lipkin and Smith, 2005), in which P represents the population, r is the growth rate, and k is the population at which the growth rate becomes close to zero.

$$\dot{P} = r \cdot P \cdot (1 - \frac{P}{K}) \tag{5.23}$$

Thus, by applying this reasoning to cells, we obtain Equation 5.24, where x_{11} is the number of cells. Here, the growth rate is $\mu(t)$, whereas the stabilized population is μ_{max}, which is obtained experimentally thanks to the absorbance values.

$$\dot{x_{11}} = \mu(t) \cdot x_{11} \cdot (1 - \frac{x_{11}}{\mu_{max}}) \tag{5.24}$$

5.1.7 *Gene circuit digital twin*

Now, by combining the sets of Equations 5.18, 5.22, and 5.24, the ODE model 5.25 is obtained, and thus the digital twin of the gene circuit.

$$\begin{cases} \dot{x_3} = C_N \cdot \frac{k_1 \cdot k_2}{d_1 + \mu} - (d_2 + \mu) \cdot x_3 \\ \dot{x_8} = C_N \cdot \frac{k_3 \cdot k_4}{d_3 + \mu} \cdot (\alpha + (1 - \alpha) \cdot \frac{(x_9)^n}{(\frac{k_{dlux} \cdot k_i \cdot C_N}{x_3})^n + (x_9)^n}) - (d_4 + \mu) \cdot x_8 \\ \dot{x_9} = D \cdot \frac{V_{cell}}{V_{ext}} \cdot x_{10} - (D + d_A + \mu) \cdot x_9 \\ \dot{x_{10}} = -D \cdot x_{11} \cdot \frac{V_{cell}}{V_{ext}} \cdot x_{10} + D \cdot \sum_{i=1}^{x_{11}} x_9 \\ \dot{x_{11}} = \mu(t) \cdot x_{11} \cdot (1 - \frac{x_{11}}{\mu_{max}}) \end{cases} \tag{5.25}$$

Thus, the final ODE model represents the change over time of the main species of our gene circuit, built in the laboratory.

5.2 MATLAB implementation

For the computational simulations of Model 5.25, MATLAB code was used (see Figure 5.2).

A set of fixed variables will stay at the same value throughout the model refinement (see Table 5.2). On the other hand, the parameters that will be estimated in the optimization process are listed in Table 5.3.

```
%x1 = LuxR
dxdt(m,1) = p.pR(k)*p.CN(k)*p.kR(k)./(p.dmR(k)+p.mu) - (p.dR(k)+p.mu)*x(m);

%x2 = GFP
c6 = p.pg(k)*p.CN(k)*p.kg(k)./( p.dmg(k)+p.mu );
Plux = x(m+2)^2./( (p.kdlux(k)*p.kd2(k)*p.CN(k)./x(m))^2 + x(m+2)^2);
dxdt(m+1,1) = c6*( p.alpha(k) + (1-p.alpha(k))*Plux )-(p.dg(k)+p.mu)*x(m+1);

%x3 = AHLint
dxdt(m+2,1) = p.D*p.Vcell./p.Vext*x(Size*Ncell+1)-p.D*x(m+2)-(p.dA(k)+ p.mu)*x(m+2);

%x4 = Number of cells
dxdt(m+3,1) = p.mu*x(m+3)*(1-x(m+3)/p.cellmax);

%AHLext
dxdt((Size*Ncell+1),1) = -p.D*x(m+3)*p.Vcell/p.Vext*x(Size*Ncell+1) +...
                          p.D*x(m+3)*sum(x(m+2:Size:Size*Ncell)) -...
                          p.dAe*x(Size*Ncell+1);
```

Figure 5.2: MATLAB implementation of the final reduced model.

Symbol	Parameter	Description	Initial value	Source
p.CN	C_N	Copy Number	High-copy: 40	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
			Low-copy: 5	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.kR	k_1	mLuxR transcription rate	2.9 min^{-1}	Rates calculator
p.pR	k_2	LuxR translation rate	4.44 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.kg	k_3	mGFP transcription rate	2 · p.kR min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.pG	k_4	GFP translation rate	0.24 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.kdlux	k_{dlux}	Half-concentration	65,000 nM · min^{-1}	Buchler et al., 2005
p.k2	k_i	LuxR-AHL dissociation rate	0.0833 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.dmR	d_1	mLuxR degradation rate	0.23104 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.dR	d_2	LuxR degradation rate	1.1552×10^{-3} min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.dmg	d_3	mGFP degradation rate	0.23104 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.dg	d_4	GFP degradation rate	1.1552×10^{-3}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.mu	μ	Dilution rate	8.6643×10^{-3} min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.alpha	α	Basal production of GFP	0.01 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.D	D	Kinetic rate of internal AHL transport	2 min^{-1}	"Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.
p.Vcell	V_{cell}	Typical volume of an *E. coli* cell	2×10^{-15} liters	Kubitschek and Friske, 1986
p.Vext	V_{ext}	Culture medium volume	2×10^{-4} liters	N/A
p.dA	d_A	Internal AHL degradation rate	4×10^{-5} min^{-1}	Pai and You, 2009
p.dAe	d_{Ae}	External AHL degradation rate	4.81×10^{-5} min^{-1}	Kaufmann et al., 2005
p.cellmax	μ_{max}	Maximum cell number	Dependant on maximum measured OD_{600}	N/A

Table 5.2: Script to parameter equivalences.

Parameter	Description
$p.CN_{L,H}$	Low-copy is estimated and high-copy is obtained as a relative number to low-copy.
$p.pG_{1,2,3,4,5}$	Five-element array for the five RBSs
$p.kdlux_{L,H}$	Two possible values depending on vector (high-copy or low-copy)
$p.dmg_{L,H}$	
$p.dg_{L,H}$	
$n_{L,H}$	
$alpha_{L,H}$	

Table 5.3: Parameters to be estimated by optimization.

Chapter 6

Phase II: Build

6.1 Introduction

All genetic circuits in this project were named according to a specific naming convention of the Synthetic Biology and BioSystems Control Lab ("Synthetic Biology and Biosystems Control Lab – SB2CL", n.d.). This is based on the creator's full name, the antibiotic resistance of the vector, and Level of each construct. In this specific case, a fourth additional element was added to the name to allow differentiation between constructions. This way, the plasmids are named based on the following structure: $pXX_1X_2XX_3XX_4$.

- **XX_1:** Initials (name and surname) of the creator.
- **X_2:** Letter representing the antibiotic resistance (C for chloramphenicol, K for kanamycin, and S for spectinomycin).
- **XX_3:** Letter representing the Level (A for α, O for Ω) and number type (1 or 2).
- **XX_4:** Vector type (origin of replication): L for low-copy and otherwise high-copy, and the RBS identifier, a number between 1 and 5.

Therefore, the plasmids created for this project are referred to as $pASSO1XX_4$ (Anna Sergeevna, Spectinomycin, and Ω1). The identifier refers to the RBS used, according to the numbering convention specified in Table 6.1.

Identifier	Ribosome binding site
1	B0030
2	B0032
3	B0034
4	J61100
5	J61101

Table 6.1: Identifiers of the and RBS in the lab.

The main objective of this project is to characterize the promoter Plux, five different RBSs, and the C_N of two types of vectors. Thus, the first step is to incorporate all necessary parts in a single cell in order to test it in an experimental environment and study its behaviour.

6.2 Construction of the genetic circuit

Dealing with inducible circuits, it is important to have a good understanding on how they work. To induce a gene circuit, we need a signal for induction. In our case, the AHL molecules bounf to LuxR protein were used to activate the promoter Plux (Lade et al., 2014)). After that, the production of the Green fluorescent protein (GFP) is started.

Now, these biological systems can be constructed in two different ways. One option is to incorporate two types of circuits into the same cell: one producing the LuxR protein needed for the Transcription factor, and another with the inducible circuit. The second possible approach is to incorporate both gene circuits in a single circuit into the cell.

There are two main reasons why the second option provides greater advantages. Firstly, by correctly selecting colonies, it can be ensured that the cells will express all the genes of interest.

Secondly, there is no need for multiple reactions to introduce both plasmids into cells, but a single step by incorporating a single plasmid that already expresses all the desired proteins (Kriz et al., 2010).

For this reason, it was decided to combine both parts (constitutive and inducible) in the same construct, resulting in a genetic circuit that expresses two different proteins in two different ways.

This circuit was constructed using six already-existing Level 1 assemblies which were combined into a vector to create a Level 2 plasmid containing both constitutive and inducible parts.

6.2.1 Constitutive circuit

For the constitutive circuit, only one Level 1 construct was designed. Synthetic Biology implies modularity, therefore the constitutive circuit was taken from a previous module named **pARKA21** (Requena Gutiérrez, 2019). The parts of this circuit are listed in Table 6.2.

Name	**Type**	**Description**	**Source**
J23106	Promoter	Constitutive promoter	Requena Gutiérrez and Ropero, 2018a
B0030	RBS	Strong RBS	Requena Gutiérrez and Ropero, 2018c
LuxR	CDS	LuxR protein coding sequence	Requena Gutiérrez, 2018b
B0015	Terminator	Transcriptional terminator	Requena Gutiérrez, 2018c

Table 6.2: Basic parts of Level 1 pARKA21 construction.

This construct, originally incorporated into an α2-type plasmid, enables the constitutive production of the LuxR protein, which will induce the production of GFP together with external AHL molecules as a TF. The result is the circuit shown in Figure 6.1.

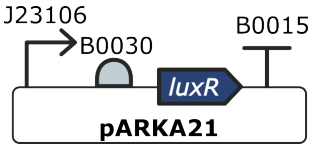


Figure 6.1: Constitutive gene circuit. LuxR is produced thanks to constitutive promoter J23106.

6.2.2 Inducible circuits for different levels of a protein

Five Level 1 constructs belonging to the **pIAKA1** collection were used (Alarcón López, 2019). The parts are listed in Table 6.3.

Name	Type	Description	Source
Plux	Promoter	Inducible promoter	Requena Gutiérrez, 2018a
B0030	RBS	Very strong RBS	Requena Gutiérrez and Ropero, 2018c
B0032	RBS	Strong RBS	Requena Gutiérrez and Ropero, 2018e
B0034	RBS	Medium strength RBS	Requena Gutiérrez and Ropero, 2018d
J61100	RBS	Weak RBS	Requena Gutiérrez and Ropero, 2018b
J61101	RBS	Very weak RBS	Requena Gutiérrez and Ropero, 2018f
GFPmut3b	CDS	GFP protein coding sequence	Requena Gutiérrez and Ropero, 2018g
B0015	Terminator	Transcriptional terminator	Requena Gutiérrez, 2018c

Table 6.3: Basic parts of Level 1 pIAKA1 constructions.

Thus, the Plux promoter combined with five different RBS sequences results in five distinct constructs. These are denoted by the common name **pIAKA1_3X1**, where X is a value between 1 and 5, following the same convention as the one shown in Table 6.1. Therefore, all these constructs share the same promoter, CDS (which codes for GFP), and terminator, where the only variable is a different sequence of RBS (Figure 6.2).

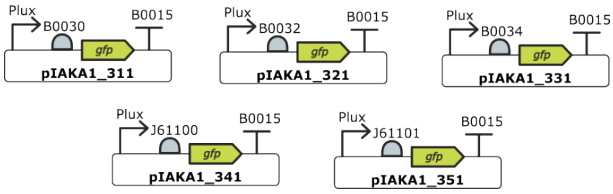


Figure 6.2: The five inducible pIAKA gene circuits. GFP is produced depending on the amount of AHL available, which binds to LuxR and induces the transcription process. The five different RBSs regulate the translation rate of each circuit.

These constructs were incorporated into an α1-type vector. It is worth noting that each RBS has a different strength, so the translation rate will vary among the constructs, being higher in those where the RBS is stronger.

6.2.3 Vector selection for final circuits

Once the parts of the final circuits in this work have been defined, it is also important to discuss the vectors into which the gene circuits are going to be incorporated.

Since the goal is to obtain a Level 2 genetic circuit, the destination vectors used were Ω type subtype 1. As one of the objectives includes the characterization of two Copy Numbers, high-copy and low-copy vectors were used. Therefore, each construct was cloned twice, once per backbone.

Thus, the destination vectors used were the ones listed in Table 6.4.

Name	Description	Source	Origin of replication
pRMS1 ω1	High-copy ω1 vector	Source	pUCOri
pIASO1S471	Low-copy ω1 vector	Alarcón López, 2019	pSC101

Table 6.4: Backbones used for the assembly of Level 2 constructs.

6.3 Final circuits

The final circuits were built by combining the pARKA21 construct with each of the constructs from the pIAKA1 collection, resulting in a total of five assemblies. However, considering that each assembly was cloned into two different destination vectors, there are 10 different circuits.

Therefore, this plasmid collection is named **pASSO1**, consisting of the five high-copy constructs shown in Figure 6.3, and the five low-copy constructs shown in Figure 6.4.

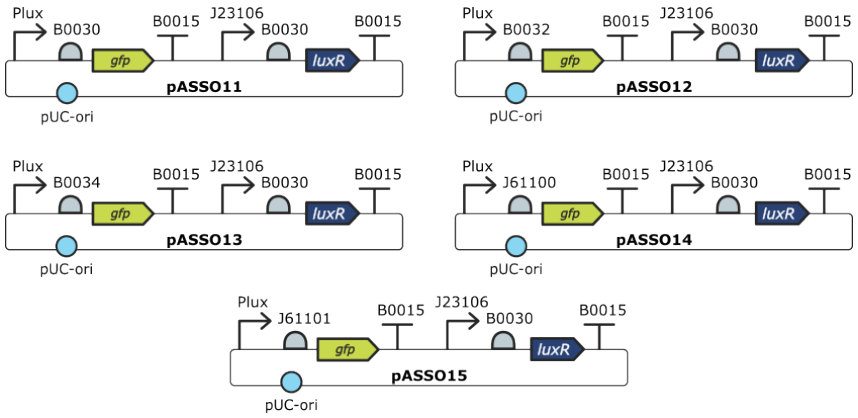


Figure 6.3: High-copy pASSO1 plasmids.

Finally, these gene circuits were transformed into *E. coli* cells as in Section 4.4.2).

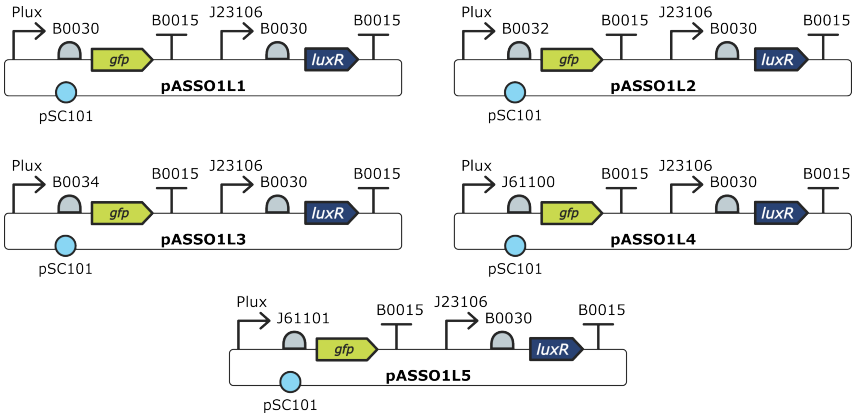


Figure 6.4: Low-copy pASSO1 plasmids.

6.3.1 Transforming final circuits into bacteria

The cells were transformed using the procotol *Transformation of chemo-competent Escherichia coli cells*, amd then seeded onto Petri as described in the *Preparation of a cell culture* protocol. After the necessary incubation period, the plates were inspected under a transilluminator with ultraviolet light.

As discussed in Section 4.4.2, the key aspect to consider is that red colonies indicate the absence of correct inserts. Additionally, it is crucial to keep in mind the expected appearance of the correct colonies based on the BioBricks that form part of their genetic circuit.

In this case, it is known that if the Golden Braid Assembly was successful and the cell incorporated the required plasmid, it should appear mostly colourless. This can be justified by the fact that the complete genetic circuit presents constitutive production of LuxR, which is colorless, and inducible production of GFP, which exhibits green fluorescence, as the name suggests. However, the production of fluorescent protein does not occur without the presence of the AHL molecule, which is necessary for induction.

By focusing on the presence of red colonies, it can be observed that in the case of high-copy constructions (Figure 6.5), some assemblies were more successful than others. Specifically, pASSO14 and pASSO15 exhibited less successful outcomes during the process. On the other hand, the low-copy circuits (see Figure 6.6) did not display a single red colony, indicating that the constructions were carried out correctly.

Now, regarding the remaining colonies, the first notable observation is that in the high-copy case, constructions pASSO11 and pASSO13 exhibit green colonies, despite the fact that, as mentioned above, these should not be producing fluorescent protein yet. However, it is also important to consider that this could be attributed to a basal production of GFP that is independent of the LuxR-AHL complex, which would make sense considering that these constructions have the strongest RBSs. Nevertheless, it is true that this result can be slightly disconcerting since pASSO12, which should have a higher translation rate than pASSO13, does not display such a high basal production.

In fact, this phenomenon is observed even in the low-copy versions of these constructions. Once again, colonies derived from both pASSO1L1 and pASSO1L3 exhibit some degree of green fluorescence, albeit to a lesser extent. This leads to consider that perhaps the theoretical assumption

that RBS B0032 has a higher translation rate than B0034 is not correct, although this will be determined once the mathematical parameters derived from the experimental data are obtained.

In any case, to confirm the correctness of the colonies before initiating any experiments, an initial preselection of four colonies per plate was performed, followed by a colony PCR (see *Colony PCRs* protocol in Section 4.4.2) of these colonies to make the final selection.

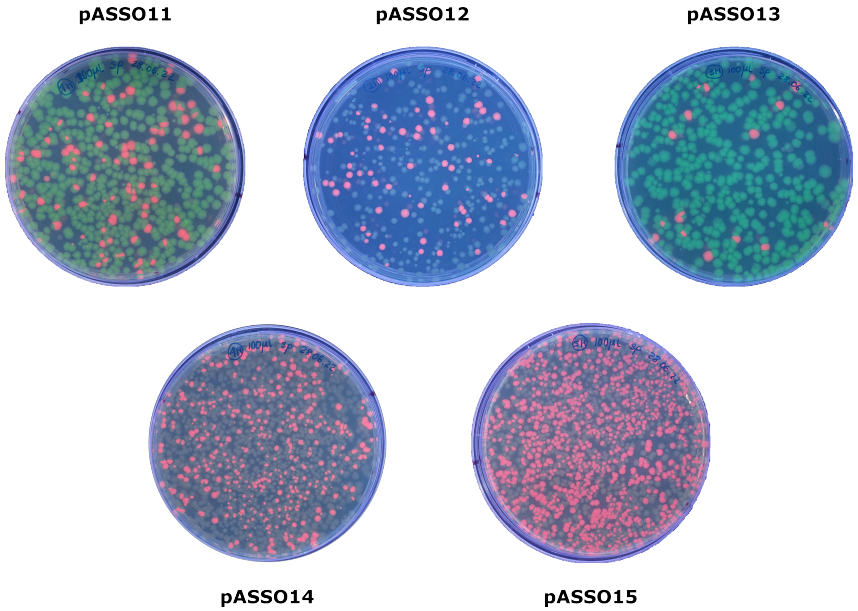


Figure 6.5: Petri dishes with grown high-copy colonies.

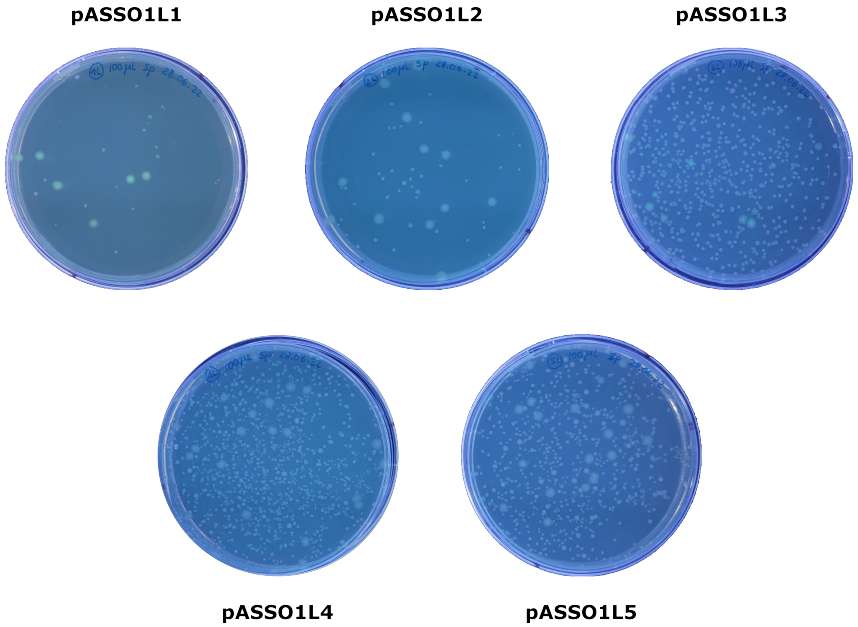


Figure 6.6: Petri dishes with grown low-copy colonies.

6.3.2 Selecting the correct bacterial colonies

For the initial preselection, three main criteria were used:

1. The colony must not be red.
2. The colony must be transparent or, alternatively, show faint green coloration.
3. The four colonies must have different morphologies in terms of colour and size.

This way, four colonies per plate were preselected and subjected to a colony PCR, which was later analyzed on an agarose gel. In this case, primers were used that delimit the Transcription Unit of the plasmids, which has a length of 2000 base pairs.

Therefore, only two colonies per gene circuit were selected based on the results obtained from the electrophoresis process, by checking the inserts that matched the expected length (see Section 4.4.2). Thus, the green-marked high-copy colonies shown in Figure 6.7 (H) and the equally marked low-copy colonies in Figure 6.7 (L) were selected.

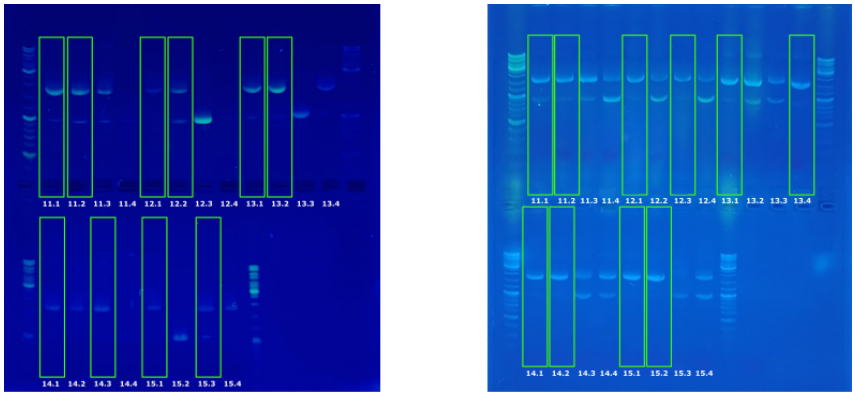


Figure 6.7: 1% Agarose electrophoresis gel showing the results of the colony PCR conducted for the fourty selected high-copy (H) and low-copy (L) colonies.

Afterwards, 5 µL of each selected colony were inoculated in 4.5 mL of LB medium with spectinomycin and grown overnight. These were then stored in glycerol following the *Glycerol stock preparation* protocol (see Section 4.4.2).

Chapter 7

Phase III: Test

7.1 Experiment set up

Once the gene circuits were transformed into the cells, they were studied by conducting several experiments using the BioTek Cytation™3 Imaging Reader (see Section 4.4.1).

The main goal of these experiments was to collect experimental data for each of the 10 gene circuits that in turn, were used to estimate some of the parameters from the Model 5.25 of the gene circuit.

More specifically, the purpose was to monitor cell growth by measuring OD_{600}, and GFP production by measuring green fluorescence in cell population.

Specifically, each experiment had a total duration of 16 hours, where at the end of the first hour, induction with AHL at different concentrations was performed.

Although an experiment lasts 16 hours, the total time required to carry each of them out is 3 days, since it is divided in four essential steps:

1. Culture preparation
2. Experimental measurements
3. AHL plate preparation and induction
4. Experimental data collection

7.1.1 Cell culture preparation

To measure OD_{600} and fluorescence, every gene circuit is growing in a liquid cell culture. This step serves the purpose of standardizing the OD_{600} of each culture, so that the experiment can begin with all the samples at the same starting point and conditions.

Thus, the steps are:

1. **Select the colonies that will be used.**

   In each experiment both high-copy and low-copy versions of the same construct were used, as well as an additional colony as a negative control, pAFSO21 (Fraile López, 2020). This control construction does not produce fluorescent protein and does not respond to AHL, allowing monitoring the proper cell growth.

2. **Growing colonies in liquid medium.**

   Once the colonies are selected, they were inoculated into 3 mL of lysogeny broth medium with spectinomycin (LB+Sp) and grown in an orbital incubator for 16 hours (overnight culture).

3. **Refresh of the colonies.**

   The next day, a refresh step is performed by taking 75 µL from each culture and inoculating them into 3 mL of M9 minimal medium with spectinomycin (M9+Sp). These cultures were incubated again in the orbital incubator and grown for another 3 hours (low-copy colonies) or 4 hours (high-copy colonies).

   To ensure they finish growing at the same time, the refresh is performed with a one-hour time difference between the two types of gene circuits (the one in high-copy and the other in low-copy). The colonies from the LB medium are now discarded.

4. **OD_{600} normalization.**

   The *Bacterial normalization* protocol was conducted by the OT-2 robot using a protocol designed in the SB2C Lab to dilute the colonies in liquid medium to an initial OD_{600} of 0.05.

   This is done to start the experiment with the same initial optical density (OD_{600}) for all cell cultures.

7.1.2 Experimental measurements

Once the cultures are diluted to the desired OD_{600}, they were distributed in a 96-well plate, filling each well with 200 µL of culture.

In each experiment, the chosen colonies are derived from the same parent construction, that is the same gene circuit in two versions: high-copy and low-copy (e.g., pASSO11 with pASSO1L1, which are high-copy and low-copy respectively).

Each sample has three replicas in rows of the 96-well plate, as shown in Figure 7.1. The first and last rows are used for M9 medium (which served as the blank) and negative control, respectively. Since the last column is used during the normalization protocol for calculating the initial OD, it is left empty.

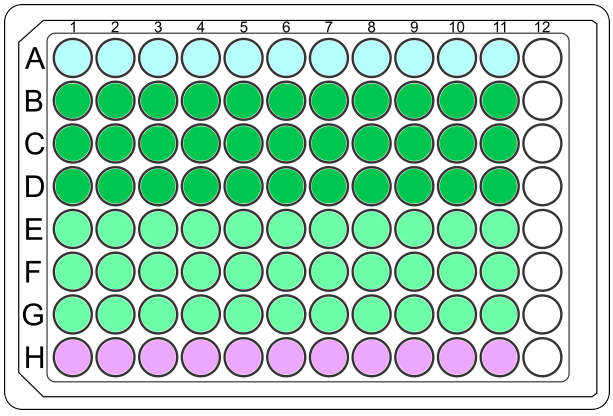


Figure 7.1: Schematic representation of the sample disposition in the 96-well plate. The colors represent the following samples: **light blue:** M9 medium; **dark green:** high-copy samples; **light green:** low-copy samples; **lilac:** negative control pAFSO21. (Template taken from Carson, 2020).

Now, it is placed in the Cytation and the protocol *Charact_ Plux_ AHL.prt* is executed. The temperature is kept constant at 37^{o}C. The protocol steps are:

1. **First step.**

   The duration is 1 hour, with continuous shaking of the plate to stimulate cell growth.

   OD_{600} and fluorescence measurements are performed every 5 minutes.

2. **Second step. (AHL induction)**

   The plate is taken out of the Cytation to allow the AHL induction.

3. **Third step.**

   The duration is 16 hours, with continuous shaking of the plate to stimulate cell growth.

   OD_{600} and fluorescence measurements were taken every 5 minutes.

7.1.3 AHL induction

During the first growth phase of the cell cultures, the AHL plate is prepared to conduct the inductions as shown in Figure 7.2.

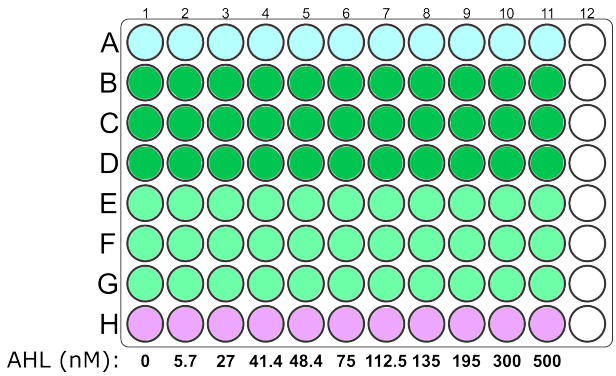


Figure 7.2: The 96-well plate with the induction concentrations in nanomolars. For each column, all wells are induced with AHL at the concentration described below it.

The purpose of performing inductions at different concentrations is to obtain a wide range of responses that will correspond to the behaviour described by the Hill function as in Chapter 5. Furthermore, it will allow the characterization of some biological parts of the gene circuits during the identification process.

Additionally, it is important to define them in a way so that the final volume added to each well does not exceed a 5% of its total volume. This is done to avoid disturbing the growing cell culture. Considering that each well has a volume of 200 µL, this means that this amount should not be greater than 10 µL.

Once the cell growth period is over, the 96-well plate will be released by the Cytation, and induction is performed by pipetting using the OT-2 robot.

When finished, the plate is put back into the Cytation and the running program is resumed, moving onto the third phase.

7.1.4 Experimental data collection

The Cytation now enters the third step. The cells will now grow for 16 hours and the OD_{600} and green fluorescence will be recorded throughout this process.

Once the experiment is completed, the plate will be released again. The data collected during experiments is stored in the Gen5 software, and thus needs to be exported to a format readable by MATLAB for further analysis (in this case, .xlsx has been chosen).

This resulted in the collection of a data set as described in Table 7.1.

Construction	**Collected data**
pASSO11	Three (3) replicas with eleven (11) levels of AHL concentration induction. OD_{600} and fluorescence measurements done every five minutes over a time-period of sixteen hours.
pASSO1L1	
pASSO12	
pASSO1L2	
pASSO13	
pASSO1L3	
pASSO14	
pASSO1L4	
pASSO15	
pASSO1L5	

Table 7.1: Description of the collected experimental data.

7.2 Experimental data analysis

To represent the experimental data obtained during the test phase, the script *Experimental_Data_Analysis.mlx* (available in the Appendix), with which data is imported and analyzed from the software integrated in the Cytation to MATLAB.

Despite having data for a total of 16 hours, and after a tidying data process, it was decided to work with the first 8 hours of each experiment. The reason for this is that cell growth shows a stabilization at this time-point, from which the system enters in a stationary state.

Now, a preliminary visualization of the experimental data was conducted. The analyzed parameters will be the OD_{600} and MEFL evolution. Although generally it is preferable to analyze the MEFL per particle because it shows the fluorescein production per protein, in this case it is visually equivalent since all inductions for each circuit showed similar growth.

7.2.1 Gene circuits pASSO11 and pASSO1L1

Beginning with circuit pASSO11, it did not exhibit the expected behaviour. For the high-copy construction, the fluorescence (Figure 7.3 B) showed abnormal results, which, when working with the Cytation, is usually due to irregular fluorescence reads.

On the other hand, its low-copy circuit (Figure 7.3, D) does appear to have grown properly, although further analysis is needed.

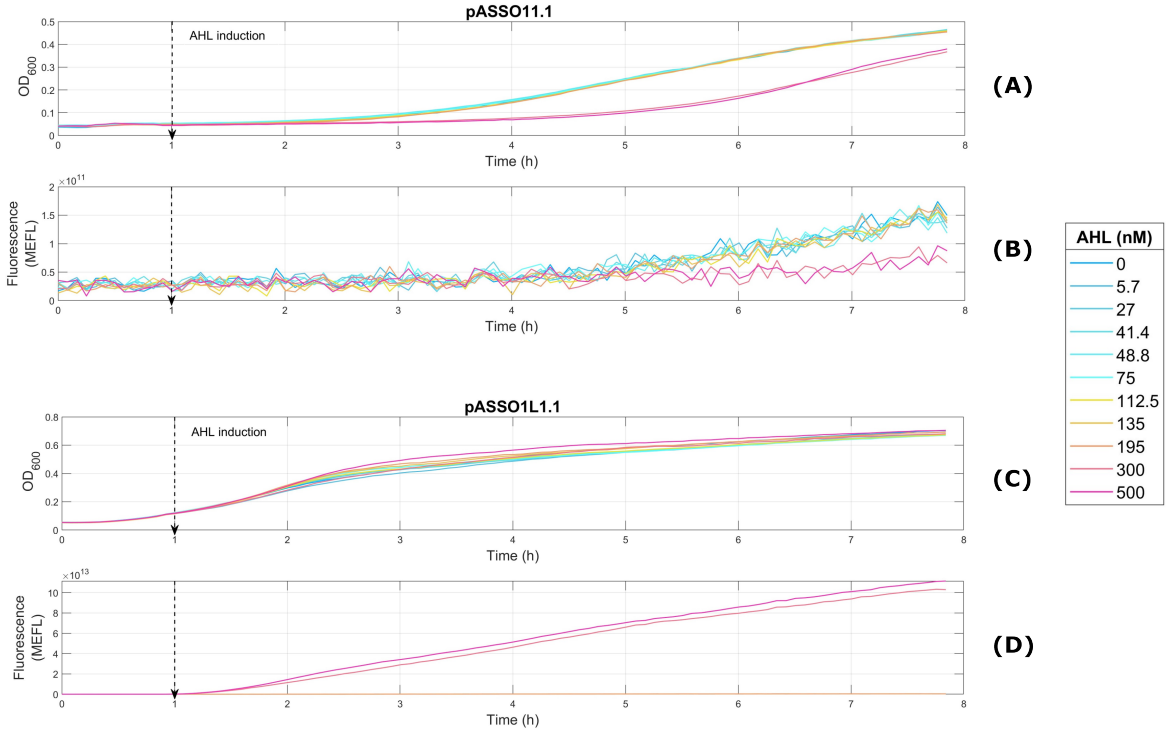


Figure 7.3: Experimental results of pASSO11, both in its high-copy and low-copy versions. The absorbance (OD_{600}) and Molecules of equivalent fluorescein (MEFL) is graphed over a period of 8 hours, which visually represents cell growth and GFP production, respectively. The discontinuous line represents the moment at which the induction with AHL was performed, which was at hour one for all experiments. Each curve is color-coded according to the AHL induction concentration used.

7.2.2 Gene circuits pASSO12 and pASSO1L2

The results for pASSO12 are shown in Figure 7.4. For this circuit, it can be seen how the initial results are consistent with the expected outcomes:

1. A practically identical growth is observed regardless of the AHL induction concentration (Figure 7.4 (A) and (C)). This is reasonable since it is known that AHL triggers the production of GFP but does not affect cell reproduction directly.
2. In contrast, when comparing the growth between the high-copy and low-copy colonies, a slightly greater dispersion is observed for the high-copy case (Figure 7.4 (A)). This is related to the fact that using high-copy plasmids is associated with higher metabolic burden. Therefore, slight differences may arise between inductions due to the higher production demand and, consequently, higher burden.
3. The Molecules of equivalent fluorescein (MEFL) show a clear relationship with the concentration of AHL. The higher the concentration is, the larger number of equivalent molecules is (Figure 7.4 (B) and (D)).
4. In the steady state, the low-copy colony exhibits lower production of GFP (Figure 7.4 (D)) in comparison to the high-copy one (Figure 7.4 (B)). This is related to the lower plasmid copy number per cell in this construction, where the replication origin has a lower Copy Number (C_N) compared to the high-copy plasmid.

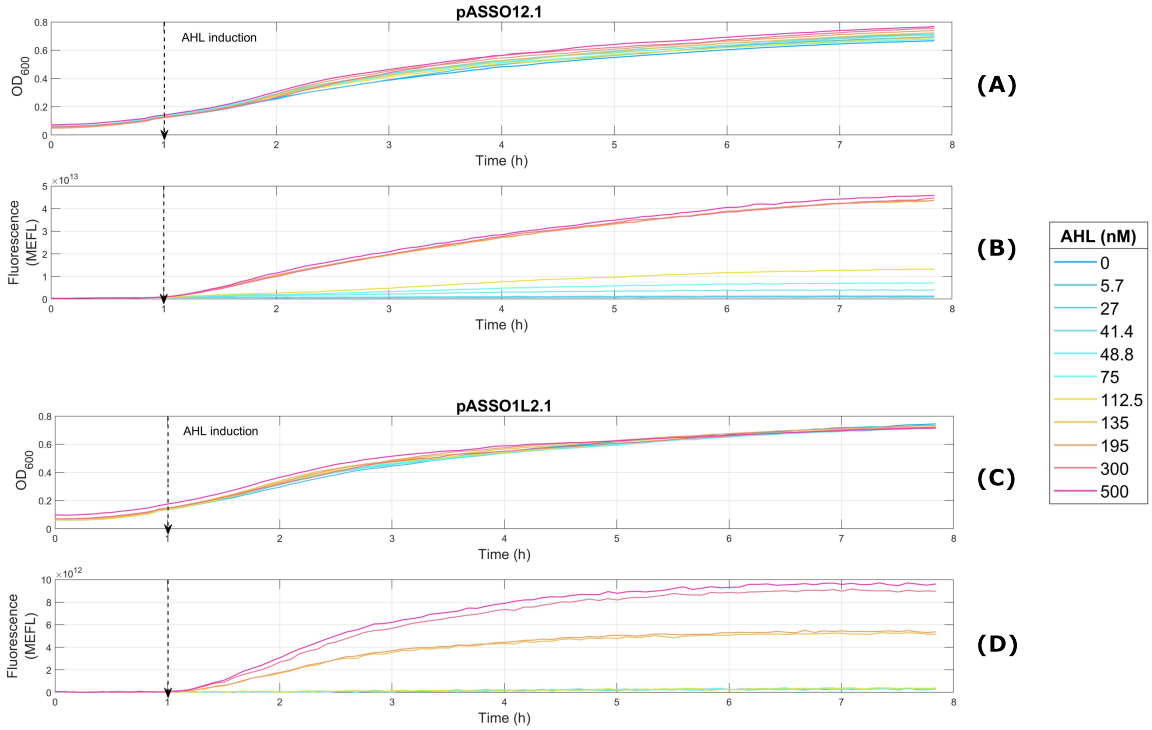


Figure 7.4: Experimental results for pASSO12, both in its high-copy and low-copy versions. The results are graphed in the same way as for circuit pASSO11.

7.2.3 Gene circuits pASSO13 and pASSO1L3

Circuit pASSO13 shows an unexpected result for the higher concentrations of induction, in which it seems like an induction at 500 nM results in a lower GFP production than at 300 nM (Figure 7.5 (B) and (D)). However, the possibility of this being due to a different cell growth can be discarded, since it can be seen how the high inductions showed almost identical growth (Figure 7.5 (A) and (C)).

Therefore, it might be due to an incorrect induction process, or to cross-contamination, which will be corroborated by checking the control colony's results in the following sections.

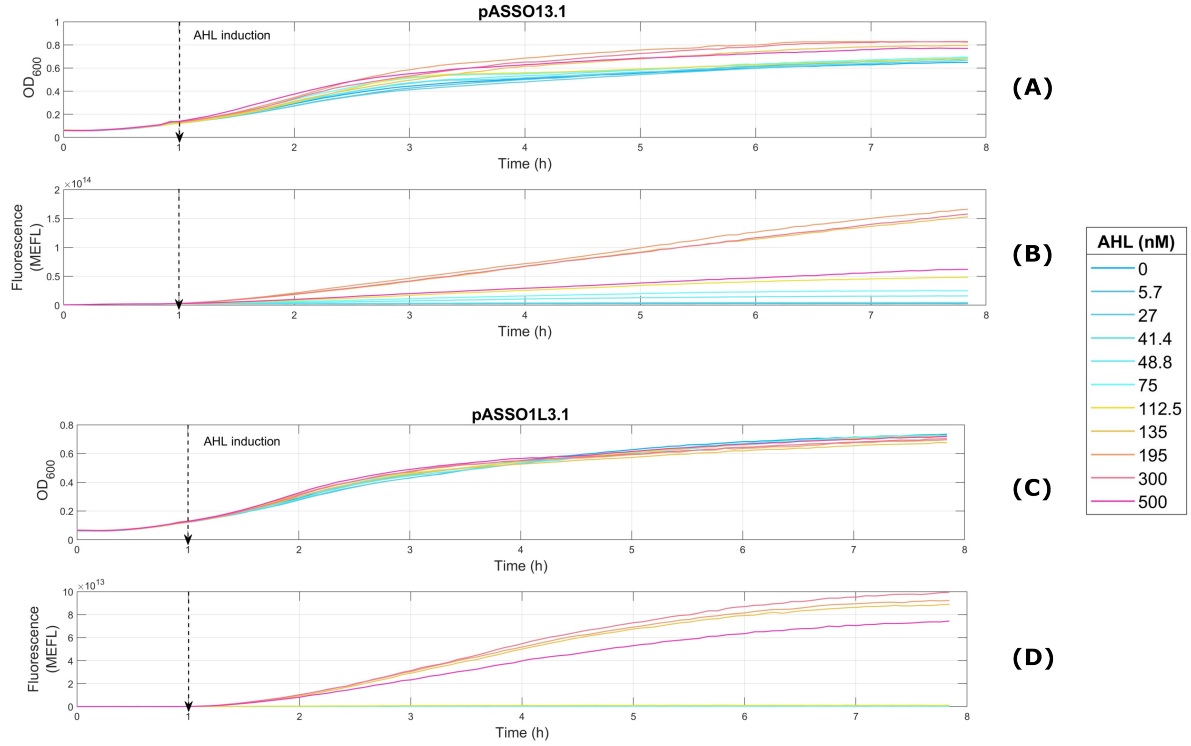


Figure 7.5: Experimental results for pASSO13, both in its high-copy and low-copy versions. The results are graphed in the same way as for circuit pASSO11.

7.2.4 Gene circuits pASSO14 and pASSO1L4

The pASSO14 (Figure 7.6) showed similar results, although the separation between the induction group of concentrations below 75 nM and those above 112.5 nM is greater than for pASSO12. This could indicate that this genetic circuit is more sensitive to drastic changes in AHL concentration when reached a certain threshold, which would also mean that the RBS is also more sensible to said variations.

7.2.5 Gene circuits pASSO15 and pASSO1L5

The high-copy pASSO15 circuit shows a similar phenomenon to pASSO13 for GFP production at 300 and 500 nM of AHL, which seem to be switched, albeit as seen in Figure 7.7 (A), it is clear that this might be due to differences in cell growth. Thus, in this case it is necessary to analyze individual protein production (MEFL/Part).

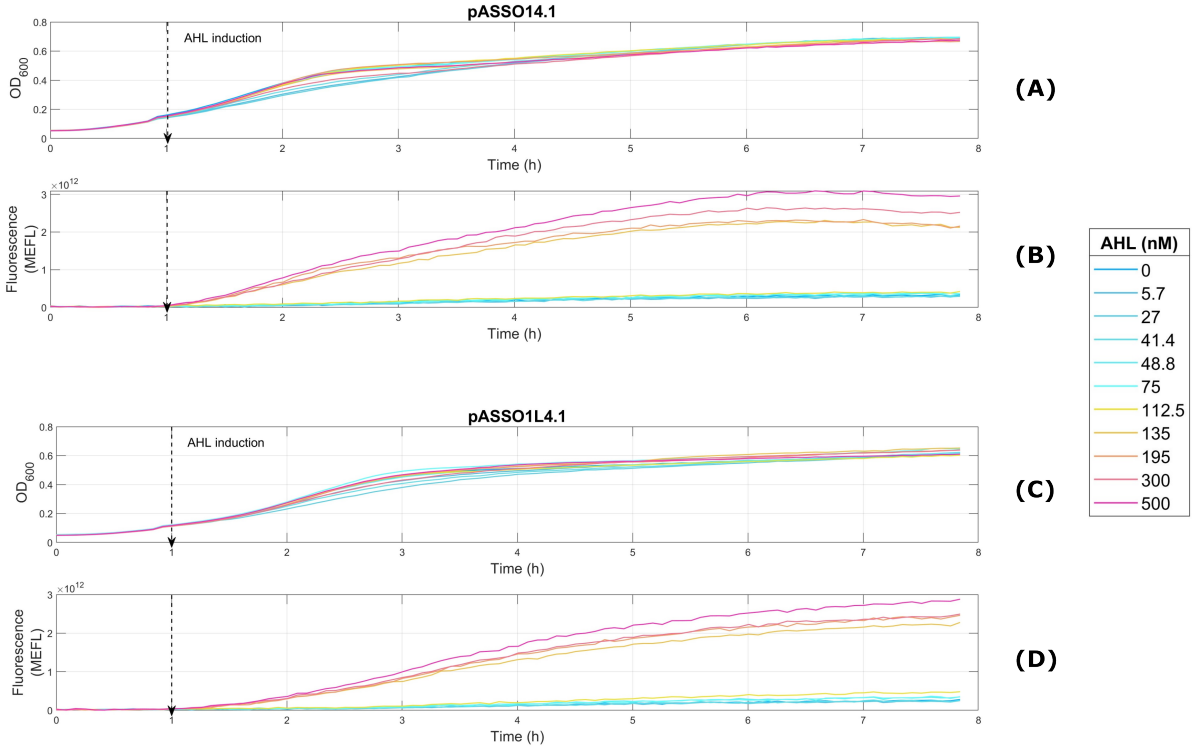


Figure 7.6: Experimental results for pASSO14, both in its high-copy and low-copy versions. The results are graphed in the same way as for circuit pASSO11.

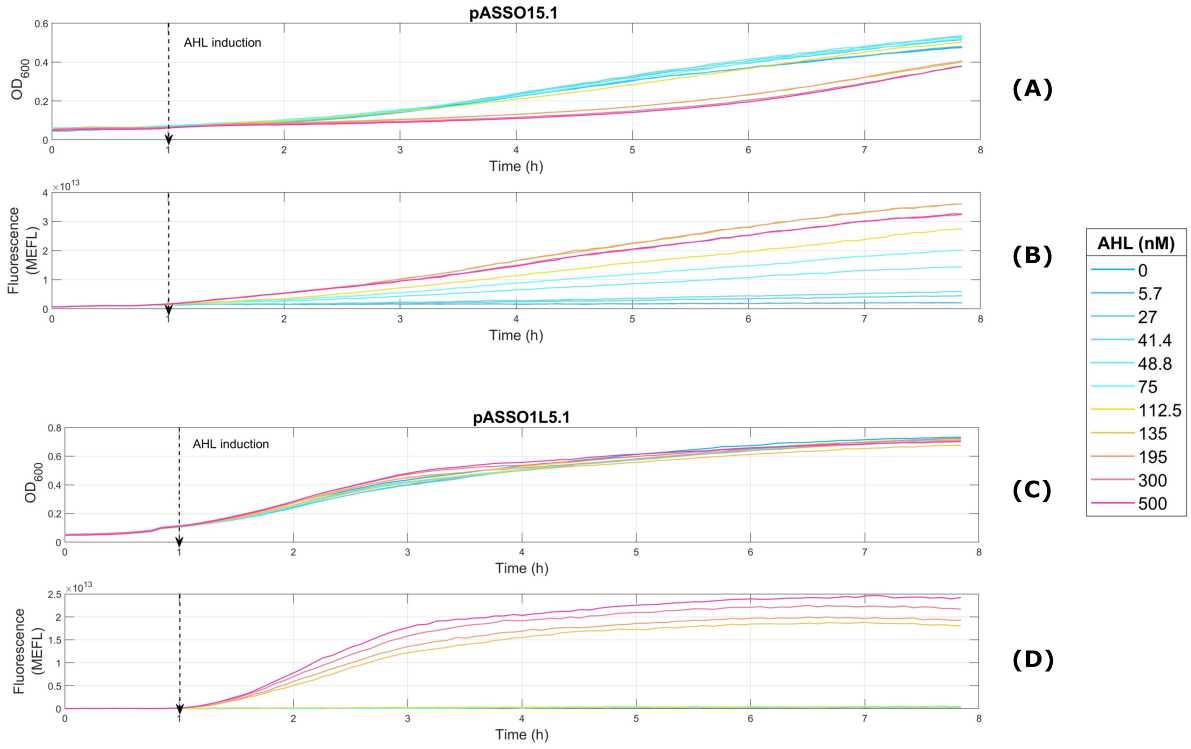


Figure 7.7: Experimental results for pASSO15, both in its high-copy and low-copy versions. The results are graphed in the same way as for circuit pASSO11.

7.3 Experimental Hill functions

From the experimental data at steady-state (end of experiment), a Hill function for each circuit was obtained.

Since pASSO11 appeared to be particularly problematic circuit, it is relevant to individually analyze its result (Figure 7.8).

On the other hand, the rest of the circuits (Figure 7.9) have demonstrated the expected behaviour. In general, there are two aspects to highlight:

1. **Higher protein production per cell in high-copy circuits.**

   The curves for both types of circuits (high-copy and low-copy) are clearly separated, being the high-copy curve above in almost every case. Thus, it is clear that these circuits present a higher protein production.

   This is expected since high-copy circuits begin with a higher DNA amount in the cell, and thus more promoters are available to be bind to.

2. **Transition in MEFL/Particles for low-copy circuits.**

The change from lower to higher production is more drastic for low-copy circuits, whereas in the high-copy case the function appears smoother.

This can be due to differences in sensibility to AHL induction. Since low-copy circuits have a lower DNA amount, it is possible that greater concentrations of AHL are required to trigger the initiation of a significant production of GFP protein.

7.3.1 Gene circuits pASSO11 and pASSO1L1

As seen in Figure 7.8, indeed the high-copy version of the pASSO11 gene circuit (high-copy version) did not respond to induction.

On the other hand, the pASSO1L1 that is thelow-copy version does seem to have a good response to the induction, but only for the 300 and 500 nM of AHL concentration. Despite this, the rest of the induction do not seem to have caused any effect, since the response levels are close to the pAFSO21 circuit, which is non-responsive to AHL.

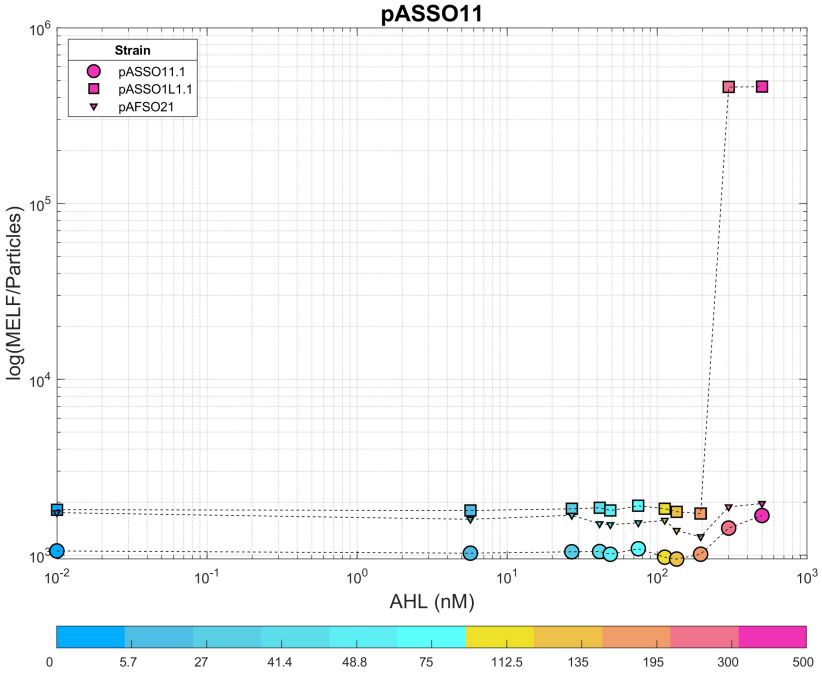


Figure 7.8: Experimental Hill function obtained for circuit pASSO11. As seen in the legend, the curve marked by the **circular** points corresponds to the high-copy version, the **square** points represent the low-copy version, and the **triangular** points – the control circuit. Note that the vertical axis is $\log_{10}$.

7.3.2 Gene circuits pASSO12 and pASSO1L2

As seen in Figure 7.9 (Left top), in circuit pASSO12 the lower concentration inductions appear very close to pAFSO21, similarly to what happens with circuit pASSO11. Nonetheless, the difference with pASSO11 resides in the fact that most AHL inductions in their higher concentration showed the expected response.

Therefore, it can not be considered with as much certainty that the inductions were not successful, but rather that the response is low to the AHL inductions for the low-copy circuits.

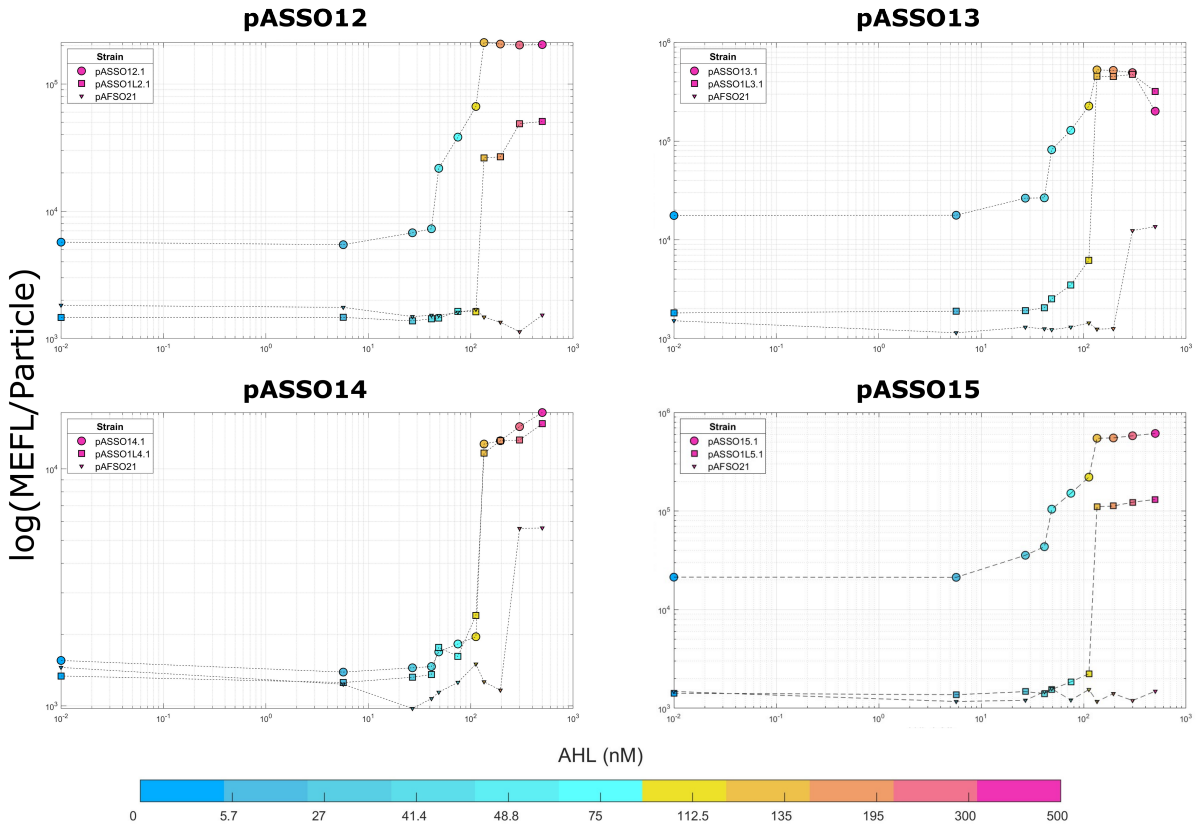


Figure 7.9: Hill functions for circuits pASSO12, pASSO13, pASSO14, and pASSO15 obtained from the collected experimental data. As seen in the legend, the curve marked by the **circular** points corresponds to the high-copy version, the **square** points represent the low-copy version, and the **triangular** points – the control circuit. Note that the vertical axis is $\log_{10}$.

7.3.3 Gene circuits pASSO13 and pASSO1L3

Circuit pASSO13 shows a slight decrease in production for higher AHL concentrations, for both high-copy and low-copy versions. Specifically, as seen in Figure 7.9 (Right top), once the AHL reaches 300 nM, the MEFL/Particle ratio starts to descend.

This may be due to a saturation because of a competition over the cellular resources. This means that when the cell reaches a certain intracellular AHL concentration, all resources are already occupied for the production of GFP, and this burden causes a decrease of GFP production.

It can also be noted how the equivalent MELF/Particle for these higher AHL concentrations slightly overlap, which could possibly indicate that, beginning from a certain concentration, the saturation point is the same independently of the Copy Number of the cell.

7.3.4 Gene circuits pASSO14 and pASSO1L4

Regarding **pASSO14** in Figure 7.9 (Left bottom), a similar phenomenon to pASSO13 is noted, albeit in this case this is also observed for lower induction concentrations.

Nevertheless, it is important to point out how in both circuits circuit pAFSO21 responded to AHL, which could indicate possible cross-contamination in both experiments.

7.3.5 Gene circuits pASSO15 and pASSO1L5

In Figure 7.9 (Right bottom), this circuit shows a similar behaviour to circuit pASSO12, where the low-copy version shows very little response to induction, being practically at the same level as control colony pAFSO21. However, as mentioned above, most of the responses to AHL inductions are the expected.

Ultimately, it appears that, in general, the circuits that did perform well in experiments do appear to have exhibited an expected response to AHL as seen in the experimental Hill functions.

7.3.6 Normalized Hill functions

Now, an important parameter is the half-concentration, at which it is considered that GFP production begins. Roughly, it is the middle point of the part of the curve that separates the lower and higher levels of productions in the Hill function.

However, in this case the aim is not to estimate this exact concentration experimentally, but rather obtain an approximation. Thus, the preliminary experimental Hill functions were normalized for every circuit, which also allows to compare the results across circuits.

These are represented in Figure 7.10. As observed, it can be seen that for most of the cases, the half-concentration ranges between AHL at 112.5 and 135 nM.

7.4 RBS Strengths

Now, it is possible to estimate the arrangement according to the strength for the four RBSs discussed. Albeit more experimental data would be needed to determine if said order is correct, we performed a preliminary estimation.

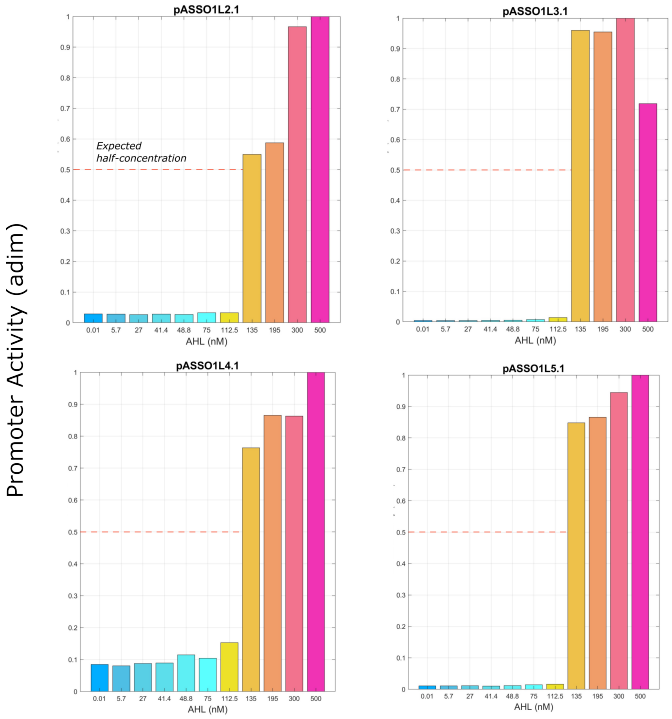
Thus, by taking into account the maximum levels of MEFL per particle produced by each circuit, the Tables 7.2 and 7.3 were obtained, for the high-copy and low-copy circuits respectively.

Construction	**High-copy (MEFL/particle)**
pASSO15	5×10^5
pASSO13	4×10^5
pASSO12	2×10^5
pASSO14	2×10^4

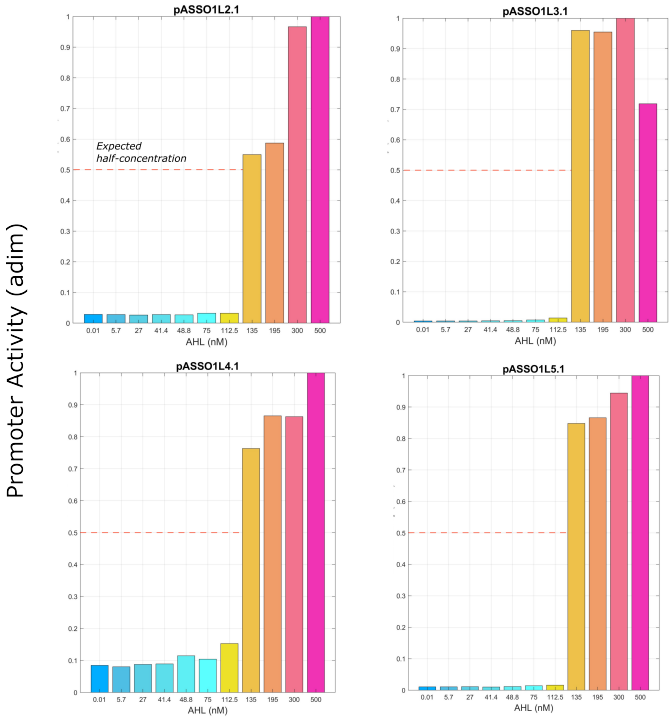
Table 7.2: Order of RBS strength based on the maximum MEFL/Particle levels produced by the high-copy circuits.

Construction	**Low-copy (MEFL/Particle)**
pASSO13	3×10^5
pASSO15	1.3×10^5
pASSO12	5×10^4
pASSO14	1.5×10^4

Table 7.3: Order of RBS strength based on the maximum MEFL/Particle levels produced by the low-copy circuits.



(a) Normalized Hill functions for the high-copy circuits.



(b) Normalized Hill functions for the low-copy circuits.

Figure 7.10: Normalized Hill functions for high-copy and low-copy circuits. Zero represents very low or null promoter activity, one represents maximum activity in relation to the rest of RBSs. The expected half-concentration is marked with a red dotted line.

Chapter 8

Phase IV: Learn

8.1 Identification of model parameters

Initially, the aim was to identify all five gene circuits, and thus use all the experimental data collected during the optimization. However, due to the invalid experimental results obtained for circuit pASSO11, it was decided not to include it in this process.

As discussed in Section 3.6, the identification process can be resumed in the steps shown in Figure 8.1.

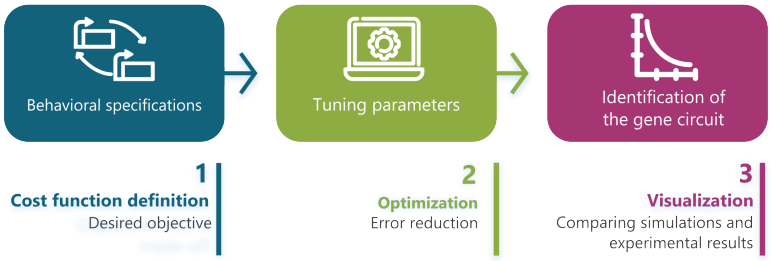


Figure 8.1: Steps for the identification of a gene circuit.

The optimization was designed in two parts: one for the high-copy circuits and another one for the low-copy. This allowed to reduce the model error, as well as better adjust model parameters representing both groups of genetic circuits.

Furthermore, it is relevant to perform this separation since it will define most of the parameters in the system (basal production, RBS behaviour, cooperativity index, GFP production rate, and degradation rates). This is due to the fact that the Copy Number defines the initial amount of DNA in the cell, and thus the initial conditions are modified as well.

8.1.1 Optimization set-up

To estimate the parameters from above, we have designed a model that will represent the Hill function, which is shown in Equation 8.1.

$$\hat{y}_k(\theta, i) = \frac{\frac{p_g^i \cdot C_N \cdot k_g}{d_m g + \mu_{max}} \cdot (\alpha + (1-\alpha) \cdot \frac{AHL_k^n}{(\frac{k_{dlux} \cdot k_{d2}}{LuxR})^n + AHL_k^n)}}{d_g + \mu_{max}} \tag{8.1}$$

Here, $\hat{y}_k(\theta, i)$ is the estimated value of the parameters in θ, which is the vector containing said values. On the other hand, i is a value from 1 to 5, depending on the RBS as in Table 6.1. This estimation will be compared with the experimental value $y_k(i)$, where k is an identifier for the AHL induction level (as described in Figure 7.2).

Note how pG^i is the translation rate of the GFP protein, and thus it will be dependant on the used RBS. Therefore, it will have four distinct values thanks to the four sets of experimental data available.

$$(\theta, i) = \begin{pmatrix} \text{kg} \\ \text{kdlux} \\ \alpha \\ \text{n} \\ (\text{pG})^i \\ \text{C}_N \\ \text{dmg} \\ \text{dg} \end{pmatrix}$$

Ultimately, the goal is to minimize cost function J, which, as described in Section 4.2.3, is the Root Mean Logarithmic Square Error (RMLSE) between experimental data and the estimated values of GFP protein. Since the interest relies on minimizing the error globally for all the parameters in each group, this error will be calculated by minimizing the RMLSE.

Thus, the resulting expression will be Equation 8.2. Notice how the error is be weighted based on the RBSs used (i, from 1 to 4) and also AHL inductions (k, from 1 up to $l = 10$).

$$\min_{\theta} J(\theta) = \frac{1}{4} \sum_{i=1}^{4} \sqrt{\frac{1}{l} \sum_{k=1}^{l} (\log_{10}(y_k(i) + 1) - \log_{10}(\hat{y}_k(\theta, i) + 1))^2} \tag{8.2}$$

This cost function was implemented in MATLAB in a script named *cost_function_pLux.m* (available in the Appendix).

Now, since two separate optimization processes were conducted, the resulting parameters were analyzed separately

8.2 Optimization results for high-copy circuits

After conducting the optimization process, the resulting parameters are listed in Table 8.1.

Parameter	Estimated value	Theoretical value	Absolute Error	RMLSE J(Θ)
kg	9.1317	5.800	3.3317	17.09%
kdlux	71549	65000	6549	
alpha HC	0.0500	0.0100	0.0400	
alpha LC	0.0567		0.0464	
n HC	3.2382	-	-	
n LC	17.9392			
pR2	0.8285	4.44	3.6115	
pR3	1.8789		2.5611	
pR4	0.0800		4.3600	
pR5	4.9980		0.5580	
CNh	52.1724	40	12.1724	
dmg	0.23350	0.23104	0.0025	
dg	4×10^{-3}	1.1552×10^{-3}	0.0024	

Table 8.1: Comparison between the estimated and theoretical parameters for the high-copy circuits.

Most of the parameters differ from their initial theoretical estimation. Albeit these previous estimations come from the literature, it must be considered the way these were originally obtained (experiments in other environments or specific scenarios, different than ours).

8.3 Optimization results for low-copy circuits

Regarding the low-copy circuits, the obtained simulated parameters for the model are listed in Table 8.2.

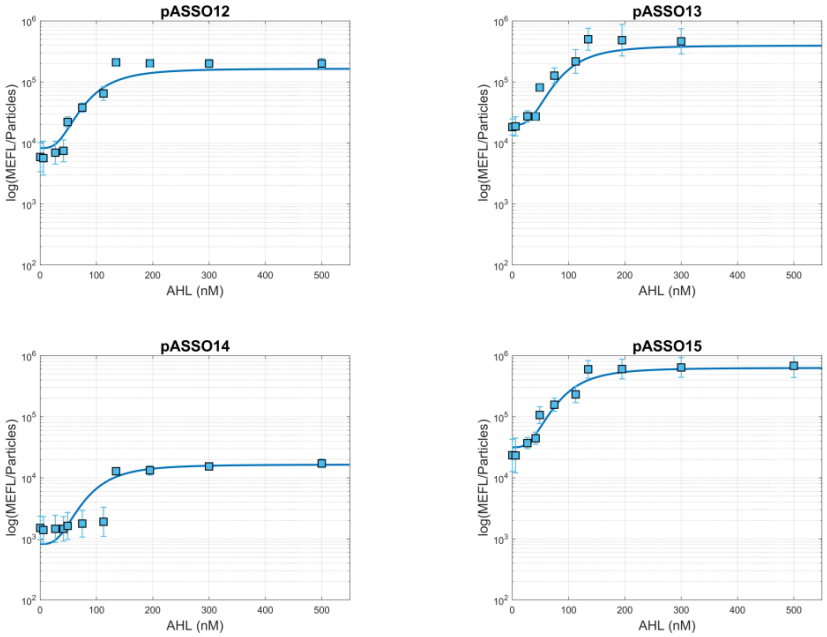
Similarly to the previous case, most of them slightly differ from their theoretical value.

Parameter	Estimated value	Theoretical value	Absolute Error	RMLSE J(Θ)
kg	3.7748	5.800	2.0252	27.64%
kdlux	56822	65000	8178	
α HC	0.0903	0.0100	0.0803	
α LC	0.0154		0.054	
n HC	6.9436	-	-	
n LC	13.5746			
pR2	2.2754	4.44	2.1646	
pR3	8.1141		3.6741	
pR4	1.1105		3.3295	
pR5	4.1261		0.3139	
CNh	8.6886	8	0.6886	
dmg	0.25670	0.23104	0.0257	
dg	5.2339×10^{-4}	1.1552×10^{-3}	6.3181×10^{-4}	

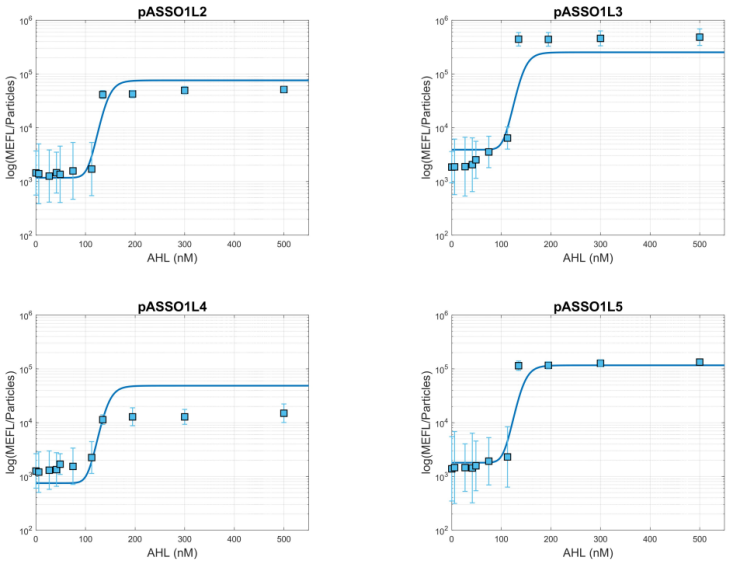
Table 8.2: Comparison between the estimated and theoretical parameters for the low-copy circuits.

8.4 Simulating Hill functions with the estimated parameters

Once the optimized parameters were obtained, these were used to simulate the new Hill functions for each high-copy (Figure 8.2a) and low-copy (Figure 8.2b) circuits.



(a) Comparison for the high-copy circuits.



(b) Comparison for the low-copy circuits.

Figure 8.2: Simulated Hill functions (continuous line) and the experimental results (square boxes).

The results show that the experimental data (marked as squared boxes in Figures 8.2a and 8.2b) are close to the estimations, there are small deviations between both experimental and simulated data, particularly for the low-copy circuits (Figure 8.2b),

Specifically, the global error of the cost function for the **high-copy** circuits model is a **17.09%**, whereas for the **low-copy** ones it is a 27.64%.

Despite these being seemingly high error values, in reality it is a good initial result. After all, several factors must be taken into account:

- **The parameters are obtained globally.** This means that, except for the GFP translation rate pG, the rest of the values are fixed for all circuits. They were not adapted or fitted to each specific circuit, but rather to their group (high-copy or low-copy) as a whole.
- **This is a living cell.** The behaviour of living organisms is not always easily predictable. Even if all experiments are conducted under the same conditions, small changes in the environment can trigger anomalous behaviours, resulting in discrepant outcomes.
- **The model or digital twin is deterministic.** In this type of models, it is assumed that the same input always produces the same output because there are no random components in the system. However, this is not always the case for living systems, where biochemical reactions are inherently random and there are many of fluctuations from the expected biological process. In our case, some of these effects have not been considered.

Chapter 9

Conclusions and Future Perspectives

9.1 Conclusions and Future Perspectives

The conclusions resulted from this project, which in turn fulfil the general objectives proposed are:

1. Firstly, the Design-Build-Test-Learn (DBTL) cycle was successfully employed to develop inducible gene circuits. At the end of each phase we can conclude:

   - **Design.** Computational design of the genetic circuits of interest was carried out to obtain a preliminary digital twin represented by the corresponding mathematical model.
   - **Build.** The ten (10) designed gene circuits were assembled in the laboratory using molecular biology techniques, and transformed into bacteria *Escherichia coli (E. coli)* for testing.
   - **Test.** Experiments were performed with the ten gene circuits to collect measurements of number of cells (Absorbance) and amount of protein produced by every circuit (Fluorescence).
   - **Learn.** The experimental data were used for the identification process to estimate the parameters of the digital twin or model of every gene circuit.

2. The basic parts of every inducible genetic circuit were designed, assembled and verified by colony PCRs. These parts were included in the laboratory's **Biological Parts Library**, so that they can be reused by other members in the design of new genetic circuits.
3. A digital twin of every gene circuit was designed to analyse its behaviour along a time window. This digital twin is a set of ordinary differential equations (ODE) that represent the biochemical reactions among the biochemical species (DNA, messenger RNA, proteins, enzymes, etc.) produced in a gene circuit.

4. The estimation of the model parameters computed the experimental data of fluorescence under different conditions of AHL induction, with the predicted data coming from the digital twin. The parameters improve the initial digital twin or model, and they explain almost the 80% of the experimental behaviour of the 10 gene circuits.

With regard to the specific objectives, all the initial target parameters were identified, i.e. those that allow characterization of the three biological parts of interest in this project:

- The inducible promoter Plux (this is, the DNA sequence responsible for the transcription process).
- The five ribosomal binding sites (RBS), which is the DNA site responsible for the translation process of a protein.
- The two vectors that define the amount of DNA (the Copy number). One vector was High-copy, and the other one Low-copy.

9.1.1 Future Perspectives

There is still work to be done to fully characterise a gene circuit. Some of the future lines derived from this project are:

1. **Re-initiation of the DBTL cycle.** Starting with a fine-tuning of the mathematical model by taking into account additional considerations such as type of nutrients added to the cell culture, or larger time periods of cell incubation. This way, we would open the possibility of starting the next iteration of the DBTL cycle.
2. **Conducting further experiments.** Conducting more number of experiments, and using a different range of induction signal (other concentrations of Acyl-homoserine-lactones AHL), would generate a more complete experimental data set. In turn, these would improve the estimation of the parameters from the digital twin.
3. **DNA Sequencing of all gene circuits.** The reason why the gene circuit pASSO11 presents unexpected behaviour might be because of mutations in the sequence. Therefore, sequencing this or even all gene circuit will provide new insights and guidelines to improve them.

Part IV

BUDGET

Chapter 10

Budget

10.1 Introduction

As an engineering project in which many computational and experimental tools and devices were used, it is important to quantify the economic value of conducting it.

The duration of this project was roughly six months, and thus this is the time-frame in which all calculations will be made.

10.2 Detailed budget

10.2.1 Labour costs

The manual laboratory work was mainly carried out by the author, who in this case is a junior engineer. A salary of €6/hour will be considered for her.

Furthermore, it is necessary to consider the positions of the supervisor and co-supervisor, both of whom are senior engineers. A salary of €12.5/hour will be considered for them.

Although laboratory technicians can also be present in this type of environment, this position will not be included since these tasks were also performed by the junior engineer, thanks to her training in molecular biology techniques.

Now, it is considered that the project had a total duration of six months, during which approximately 15 days of joint work between the supervisors and the student were necessary for instruction and learning. Thus, a total of 1,056 hours is allocated to the student, and 120 hours each to the supervisors.

In this way, the costs associated with labor are obtained as shown in Table 10.1. The total labour cost of this project amounts to **nine thousand three hundred thirty-six euros**.

Position	Unit cost (€/hour)	Amount (h)	Total cost (€)
Junior Engineer	6	1,056	6,336
Senior Engineer	12.5	240	3,000
Total			9,336

Table 10.1: Labour costs.

10.2.2 Hardware and software costs

The costs associated with hardware and software licenses used were not considered by means of their market price, but rather measured in terms of the amortization factor (A.F) over the project's duration time (6 months).

In this way, the hardware and software used are described in Table 10.2, along with their corresponding prices and amortization factors. It is considered that the laptop has a lifespan of approximately 7 years (84 months), while the licenses are amortized over a period of 12 months.

Thus, these components amount to a total of **one hundred eighty-four euros and fourteen cents**.

Component	Cost (€)	Amount (u)	A.F	Total cost (€)
Dell Laptop	800.00	1	6/84	57.14
MATLAB R2022a	250.00	1	6/12	125.00
Overleaf	0.00	1	6/12	0.00
Benchling	0.00	1	6/12	0.00
BioTek Gen5	Included with Cytation	1	6/12	N/A
Total				184.14

Table 10.2: Hardware and software costs.

10.2.3 Materials and instruments costs

The budget associated with materials and instrumentation will be considered in the same way as the used hardware, albeit considering a lifespan of 10 years (120 months). Additionally, an initial material purchase (pipette tips, Eppendorf tubes, etc.) with a value of €600 has been considered, and it is assumed that the entirety of it has been used throughout the project.

Therefore, the costs associated with the materials described in Section 4.4.1 are presented in Table 10.3, amounting to a total of **five thousand six hundred seventy-four euros and one cent**.

Instrument	Cost (€)	Amount (u)	A.F	Total cost (€)
Initial Material stock	600.00	1	N/A	600.00
TECHNÉ 3Prime Thermal Cycler	3,425.68	1	6/120	171.28
Opentrons OT-2 Robot	15,569.96	1	6/120	778.49
BioTek Cytation 3 Imaging Reader	75,000.00	1	6/120	3750.00
Stuart SI500 Orbital Shaker Incubator	4,980.62	1	6/120	249.03
Memmert IN110 Incubator	2,504.00	1	6/120	125.20
Total				5,674.01

Table 10.3: Materials and instruments cost.

10.2.4 Reaction costs

Now, it is also important to include the cost of the GB Assembly, as well as of the colony PCRs. In the first case, 10 gene circuits were assembled, and thus that is the amount of GB reactions conducted. Regarding the colony PCRs, four colonies per gene circuit were selected to conduct this reaction, and therefore 40 PCRs are considered.

The cost of said reactions was calculated based on the reactants used to conduct them, as detailed in Section 4.4.2, obtaining Table 10.4. This way, the reaction costs amount to a total of **eight hundred ninety-four euros**.

Reaction	Cost (€)	Amount (u)	Total cost (€)
Golden Braid Assembly	28.60	10	286.00
Colony PCR	15.20	40	608.00
Total			894.00

Table 10.4: Reaction costs

10.3 Total budget

10.3.1 Material execution budget

Considering all previous sections, the total material execution budget amounts to **sixteen thousand eighty-eight euros and fifteen cents**.

10.3.2 Contracted operation budget

According to the General Regulations of the Law on Public Administration Contracts, approved by the Royal Decree 1098/2001 (Ministerio de Hacienda, 2001), additional costs associated to the indirect costs and industrial benefit must be added to obtain the total budget. These are calculated by taking a 13% and a 6% of the material execution budget, respectively. An additional 21% in terms of Value Added Tax (VAT) must be added as well.

Thus, the resulting budget ascends to **twenty-three thousand one hundred sixty-five euros and thirty-three cents**, as per Table 10.5.

Denomination	**Cost (€)**
Material execution budget	16,088.15
Indirect costs	2,091.46
Industrial benefit	965,29
Total before VAT	19,144.90
VAT	4,020.43
Contracted operation budget	23,165.33

Table 10.5: Total budget, including before and after VAT.

Appendix A

Conference papers derived from this work

CEA comité español de automática

XLIV Jornadas de Automática 2023

Sesion: <P>

XLIV Jornadas de AUTOMÁTICA 2023 — ZARAGOZA

https://jautomatica.es

Towards automation of the Design-Build-Test-Learn (DBTL) bioengineering cycle: Application to the testing and characterization of standard bioparts.

Pushkareva, Anna[a], Beltran, Jaime[a], Díaz-Iza, Harold[a], Arboleda-García, Andrés[a], Boada, Yadira[a], Vignoni, Alejandro[a,*], Picó, Jesús[a]

[a]*Systems Biology and Biosystems Control Lab, Instituto de Automática e Informática Industrial, Universitat Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España.*

To cite this article: Pushkareva, Anna, Beltran, Jaime, Díaz-Iza, Harold, Arboleda-García, Andrés, Boada, Yadira, Vignoni, Alejandro, Picó, Jesús. 2023. Towards the automation of the Design-Build-Test-Learn (DBTL) bioengineering cycle.
XLIV Jornadas de Automática 00, 1-5. https://doi.org/

Resumen

El ciclo Diseño-Construcción-Prueba-Aprendizaje (DBTL) es un marco crucial en Biología Sintética para el desarrollo y optimización de sistemas biológicos. Sin embargo, la naturaleza manual del ciclo plantea limitaciones en términos de tiempo y mano de obra. Este artículo se centra en la aplicación de técnicas de automatización al ciclo DBTL, concretamente en el ensayo y caracterización de biopartes estándar, que son componentes esenciales de los circuitos genéticos. La automatización del proceso de ensayo puede mejorar significativamente el rendimiento, la fiabilidad y la reproducibilidad. En este artículo se analizan los retos asociados a los métodos de ensayo manuales y se exploran diversas estrategias y tecnologías de automatización que pueden resolverlos. Los métodos de cribado de alto rendimiento, la robótica de laboratorio y los algoritmos de análisis de datos son elementos clave en el proceso de automatización. Se examinan estudios de casos y avances recientes en la automatización del ciclo DBTL para pruebas de biopartes. La integración de la automatización en el ciclo DBTL ofrece numerosas ventajas, como una mayor eficacia, estandarización y control de calidad de las biopartes. También permite la exploración de espacios de diseño más amplios y la creación rápida de prototipos de sistemas genéticos complejos. Este artículo ofrece una revisión exhaustiva del estado actual de la técnica y las perspectivas de futuro en la automatización del ciclo DBTL para el ensayo y la caracterización de biopartes estándar.

Palabras clave: Automatización, Diseño-Construcción-Prueba-Aprendizaje, Estimación de parámetros, Opentrons, Circuitos genéticos.

Hacia la automatización del ciclo de bioingeniería Diseño-Construcción-Prueba-Aprendizaje (DBTL): Aplicación al ensayo y caracterización de biopartes estándar.

Abstract

The Design-Build-Test-Learn (DBTL) cycle is a crucial framework in Synthetic Biology for the development and optimization of biological systems. However, the manual nature of the cycle poses limitations in terms of time and labor. This paper focuses on the application of automation techniques to the DBTL cycle, specifically in the testing and characterization of standard bioparts, which are essential components of genetic circuits. By automating the testing process, throughput, reliability, and reproducibility can be significantly improved. This paper discusses the challenges associated with manual testing methods and explores various automation strategies and technologies that can address these challenges. High-throughput screening methods, laboratory robotics, and data analysis algorithms are key elements in the automation process. Case studies and recent advancements in automating the DBTL cycle for biopart testing are examined. The integration of automation in the DBTL cycle offers numerous advantages, including increased efficiency, standardization, and quality control of bioparts. It also enables the exploration of larger design spaces and rapid prototyping of complex genetic systems. This paper provides a comprehensive review of the current state-of-the-art and future prospects in automating the DBTL cycle for testing and characterizing standard bioparts.

Keywords: Automation, Design-Build-Test-Learn, Parameter estimation, Opentrons, Genetic circuits.

1. Introduction

In the field of Synthetic Biology, the Design-Build-Test-Learn (DBTL) cycle (Figure 1) serves as a fundamental framework for the development and optimization of biological systems (Cummins et al., 2023). This iterative process involves designing genetic constructs, building them through molecular biology techniques, testing their functionality, and learning from the obtained results to refine future designs. The DBTL cycle has been pivotal in advancing synthetic biology and enabling the engineering of living organisms with desired traits and functions (Tellechea-Luzardo et al., 2022; Carbonell et al., 2020).

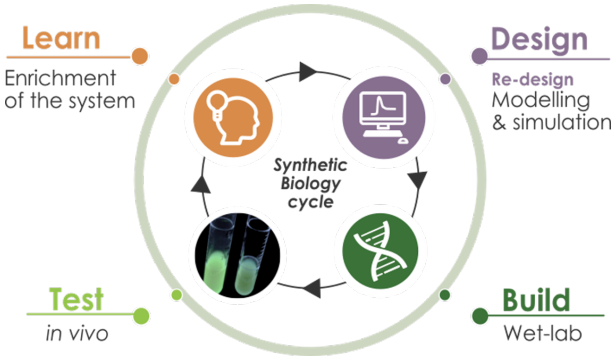


Figure 1: Design, Build, Test and Learn bioengineering cycle used in Synthetic Biology.

While the DBTL cycle has proven its effectiveness in the design and construction of biological systems (Gurdo et al., 2023), it is a labor-intensive and time-consuming process that often requires significant manual intervention. As the complexity of bioengineering projects continues to increase, there is a growing need to streamline and automate the DBTL cycle to accelerate the development of biological solutions (Cummins et al., 2023).

There is work being done towards a fully automated DBTL cycle tackling one of the steps at the time; mainly for the automation of the design step (Buecherl and Myers, 2022; Radivojević et al., 2020; Vidal et al., 2022b; Boada et al., 2021, 2022a,b), build step (Ko et al., 2022; Kang et al., 2022; Bryant Jr et al., 2023), for the test step there are advances in the calibration of the measurements (Beal et al., 2022; González-Cebrián et al., 2023), and in the automation of wetlab protocols in general(Vidal-Peña, 2023). Finally for the learn step (Vidal et al., 2022a; Yanez Feliu et al., 2020; Boada et al., 2019b). However, there are not many examples of automation of the test and learn steps combined. This paper aims to explore the application of automation techniques to the DBTL cycle steps Test and Learn, specifically focusing on the testing and characterization of standard bioparts. Standard bioparts, such as promoters, terminators, and coding sequences, form the building blocks of genetic circuits and are widely used in bioengineering projects. Efficiently testing and characterizing these bioparts is crucial for their reliable integration into larger genetic systems and the predictable behavior of engineered organisms (Boada et al., 2019a).

Automation has the potential to revolutionize the testing and characterization of standard bioparts by improving throughput, reliability, and reproducibility (Cummins et al., 2023). By leveraging advanced laboratory robotics, high-throughput screening methods, and data analysis algorithms, automation can significantly accelerate the iterative testing process and provide valuable insights for bioengineers. Furthermore, automation can enhance the standardization and quality control of bioparts (Buecherl and Myers, 2022), ensuring their compatibility and reliability across different projects and laboratories (Beal et al., 2020).

This paper will discuss the current challenges and limitations of manual testing and characterization methods for standard bioparts. It will then explore various automation strategies and technologies that can be employed to address these challenges. Additionally, we will examine with a case study the proposed advancements in the automation of the DBTL cycle for the testing and characterization of standard bioparts.

2. Automation of the Test step

This work is focused on the Test and Learn steps of the DBTL cycle, so the starting situation is where there exist several genetic constructs in the lab, and we need to test them and learn from these experiment to improve the models and characterize the used bioparts. First we deal with the Automation of the Test step. For this we implemented a automated protocol combining the Agilent Biotek plate reader (Figure 2) with the Opentrons OT-2 (Figure 3).

Figure 2: Agilent Biotek Cytation 3 plate reader. This plate reader allows us for incubation at 37ºC, agitation and measurement of both absorbance and fluorescence.

This workflow implements the protocol for the

*Corresponding Author: vignoni@isa.upv.es

normalization of initial concentrations of 7 bacterial culture samples for a 16 hour incubation/measurement experiment. Specifically, the protocol is divided into two parts, the first is a 1:4 dilution of the culture in the culture medium Minimal Media M9 salts plus 20 % of glucose (blank), this dilution will be measured and the optical density (OD) of the 7 samples will be taken so that in the second part of the protocol all the culture are normalized to an OD of 0.1. Later on, this dilutions are distributed into the measurement 96 well plate. To carry out this whole process it is necessary to have a specific configuration of the OT2 and OT-2 deck (Figure 4), to make use of the Agilent Biotek Cytation measurement program and to use the Jupyter notebook linked to the OT2 to run the protocol.

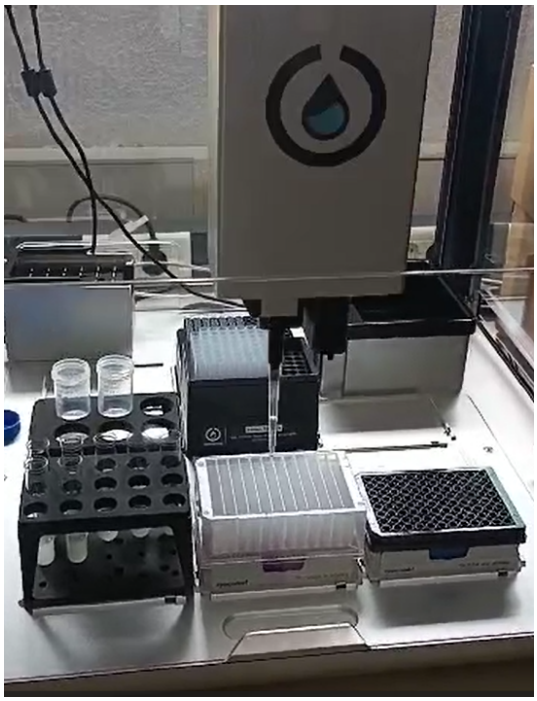


Figure 3: Opentrons OT-2 liquid-handling robot. The OT-2 already prepared to execute the protocol with all the labware in the right position. Pipettes used are the Multichannel P300 in the right mount, and the Single Channel P1000 in the left mount.

2.1. Procedure

1. Set up the OT2 as shown in the Figure 3. Pipettes used are the Multichannel P300 in the right mount, and the Single Channel P1000 in the left mount.
2. Set up the deck as follows (also shown in Figure 4):
   SLOT 1: Opentrons 15 tube rack with falcon 14 ml round
   SLOT 2: Porvair 96 deep well plate 2ml conical
   SLOT 3: Genier bio coldblock 96 well plate 400ul
   SLOT 4: Opentrons 6 tube rackwith falcon 50 ml conical
   SLOT 8: Opentrons 96 tip rack 1000ul
   SLOT 11: Opentrons 96 tip rack 300ul

3. In the third step the different liquids that the OT2 will handle will be added. The order in which the samples are placed is of great importance for the correct functioning of the protocol, in this case we have chosen the sequence A1, A2, A3, A4, A5, B1 and B2 (Figure 4).

4. The fourth step is the calibration and adjustment of the offsets. Calibration is performed by loading a python script with a small protocol in the OT2 APP with the necessary labware previously loaded. Once this protocol is selected the user has only to run the Labware Position Check to start the manual calibration. Once the offsets are obtained, do not close the window with the data, because they have to be copied to each of the protocols to be executed in Jupyter (Figure 5).

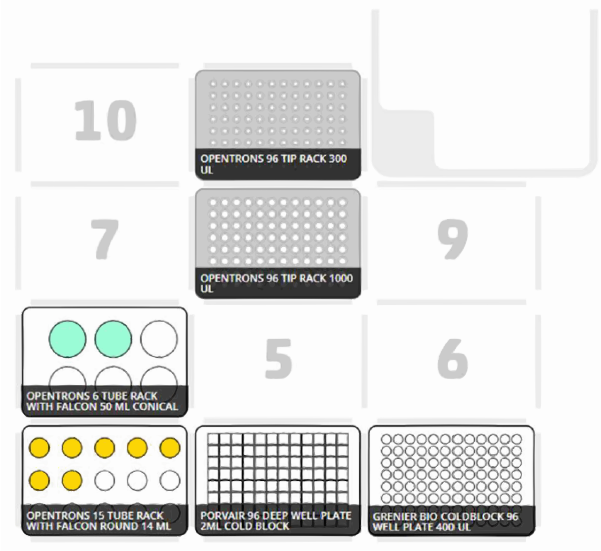


Figure 4: Deck of the Opentrons OT-2 liquid-handling robot prepared for the protocol.

5. Execution of the first part of the protocol from Jupyter. For this it is necessary to change the variable action to 2 of the protocol, to start executing the code and then launch the code by pressing Ctrl+Enter.

6. Once the whole process has been completed, the SLOT 3 plate will carry the samples in the 12 column, and it must be transfered to the plate reader to perform the Absorbance measurement. The OD measurements obtained should be saved into the provided Template Spreadsheet as a comma separated values files. This template calculates the desired dilutions and volumes to be used later by the OT-2.

7. Upload the Template csv files with the measured values to the OT-2 Jupyter environment.

8. Execution of the second part of the protocol as done in the fourth step by putting back the 96 well plate to SLOT 3 without the lid, and adding the offsets and changing the action variable to 2 of the protocol to start executing the code and then launch the code by pressing Ctrl+Etner.

Figure 5: Using a simple protocol with the OT-2 APP allows us to get the calibration values for each SLOT.

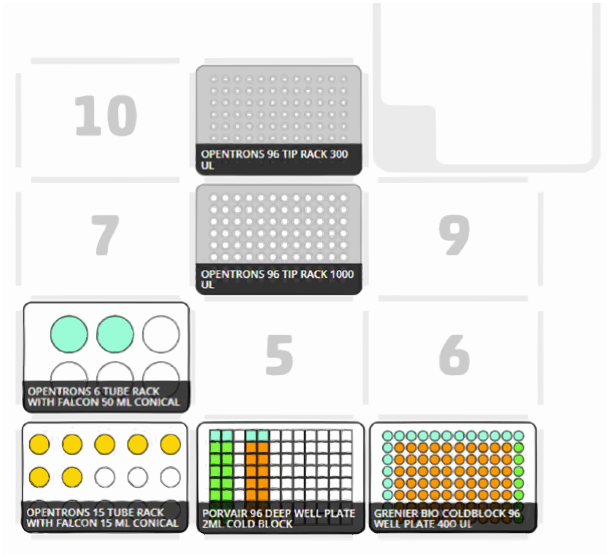


Figure 6: Deck of the Opentrons OT-2 liquid-handling robot after the completion of the two protocols. The sample 96 well-plate is prepared for growth/measurement experiment.

The use of this protocol establishes the systematic preparation of the 96 well-plate for the growth/measurement experiment, by ensuring a constant initial concentration of cultures across the plate.

Once the experiment is done, we have a big dataset of absorbance and fluorescent measurements from the selected sample.

3. Using data in the Learn step

Using the previously explained workflow for the Test step of the DBTL cycle, we obtain a dataset of measurements for the two genetic circuits shown in Figure 7.

We have 10 technical replicates in 4 experiments performed on different days for each one of the devices. While incubating at 37^{o}C and 230 rpm in a high-speed double orbital shaking, the absorbance was recorded at 600nm, and the fluorescence was measured at 530nm with an excitation of 488nm for 16 hours. The calibration of both absorbance and fluorescence was done using our calibration protocol (Beal et al., 2022; González-Cebrián et al., 2023). In Figure 8, the replicate-averaged data show: the number of bacteria (Particles), the number of molecules (MEFL), the number of molecules per cell (MEFL/Part), and the growth rate calculated with an anti-causal filter in Matlab (filtfilt command).

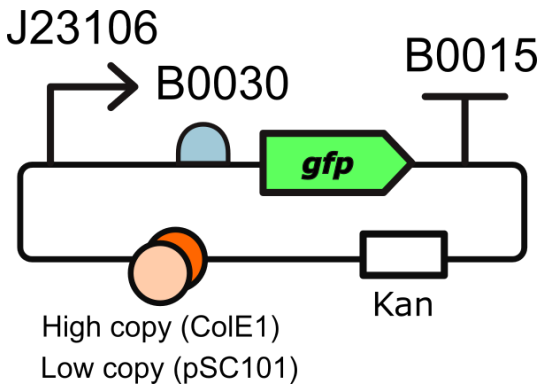


Figure 7: A constitutively expressed GFPmut3b in a high copy plasmid (ColE1 ori) and a low copy plasmid (pSC101 ori) both with the same RBS (BBa_B0030) and Promoter (BBa_J23106).

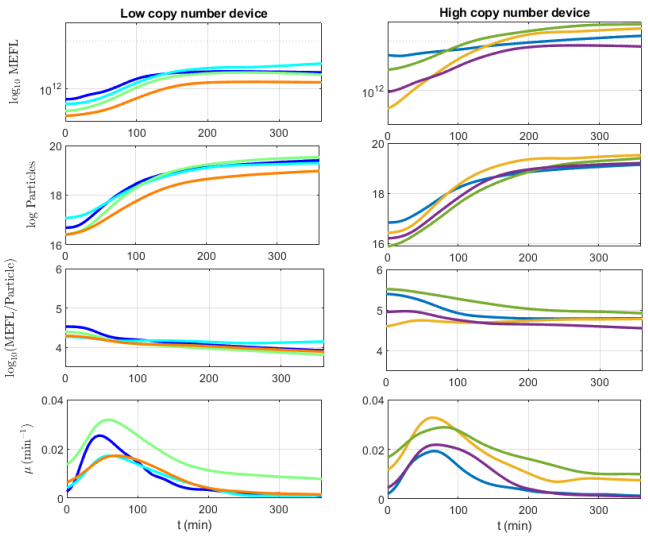


Figure 8: Experimental data for the two devices obtained using the automated workflow for the Test step. From up to bottom: Number of bacteria (Particles), Number of molecules (MEFL), number of molecules per cell (MEFL/Part), and growth rate. Left, low copy number device, right: high copy number device.

With this dataset, we perform a parameter identification by taking a simple growth-independent protein production model:

$$\frac{dN}{dt} = \mu(t)N \tag{1}$$

$$\frac{dG}{dt} = \frac{C_N\, k_G\, p_G}{dm_G + \mu_{max}} - d_G G - \mu_{max} G \tag{2}$$

where $\mu(t)$ is the growth rate, N is the number of particles, C_N is the copy number, k_G is the transcription rate of the protein, p_G is the translation rate of the protein, dm_G is the dilution rate of mRNA, d_G is the dilution rate of the protein, μ_{max} is the maximal growth rate, and G is the protein concentration. In the first place, we assume as five the value for the low copy number construct ($C_N = 5$), then we perform the identification obtaining the parametric values shown in the first row of Table 1 using the Genetic Algorithm of MathWorks (2023). After that, we just evaluate the same cost function assuming that the only change from the low copy number device to the high copy number one is precisely the copy number, from 5 in the case of the low copy to 35 in the case of the high copy (Morgan, 2023).

In particular from the dataset, we take a $\mu_{\max} = 0.02321\ [min^{-1}]$ for both low and high copy. By doing this, we can use the same identified parameter values from the low copy number device to predict the protein production of the high copy number device. As shown in Table 2, we obtain a similar error in prediction and optimization.

Table 1: Estimated parameters for Low copy number device.

	k_G	p_G	d_{mG}	d_G	C_N
Low copy	9.91	2.18	0.36	0.0021	5
High copy	9.91	2.18	0.36	0.0021	35

In Figure 9, the experimental data for each device is shown together with the optimized model for the low copy number device, which was used to obtain values for the parameter. In addition, the figure shows the predicted protein production for the high copy number device (using the low copy number parameters with a change in the copy number).

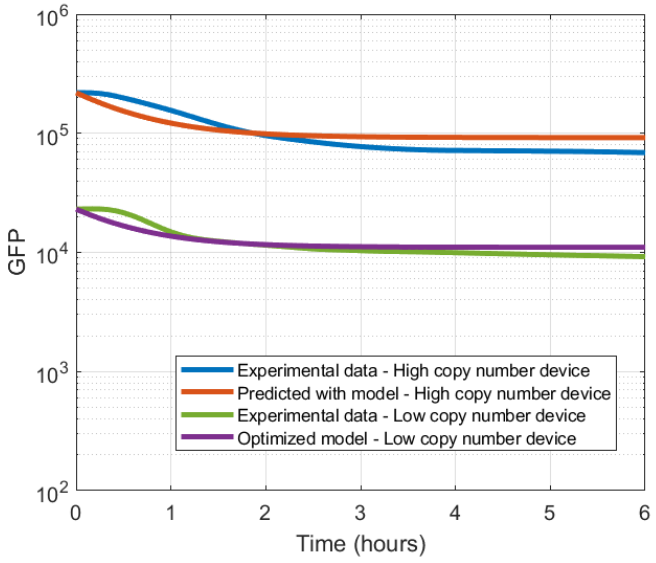


Figure 9: Optimized and predicted protein production for the low copy number device and high copy number device respectively.

Table 2: Prediction errors

	MSE	RSE
Low copy (optimization)	3.5005e+08	4.0695
High copy (prediction)	3.4719e+08	3.1733

4. Conclusions

This paper proposes an automation of the Test step of the DBTL and using the experimental data obtained for characterization of standard bioparts. With a simple model of protein production and the experimental data we can obtain parameter values that allows us to accurately predict the protein production level of another device only using the maximal growth rate of that construct. This paves the road for modeling complex devices and using these models to obtain sensible prediction of their outputs. The integration of automation into the DBTL cycle holds great promise for advancing the field of bioengineering and synthetic biology. It not only increases the efficiency and reliability of biopart testing but also enables the exploration of larger design spaces and the rapid prototyping of complex genetic systems. Ultimately, automation can facilitate the development of novel bioengineered solutions with enhanced functionality and applicability in areas such as healthcare, biomanufacturing, and environmental sustainability. By embracing automation, we can overcome the limitations of manual approaches and unlock the full potential of bioengineering to address pressing societal challenges and pave the way for innovative biological solutions.

Acknowledgments

This research was funded by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 grants number PID2020-117271RB-C21, TED2021-131049B-I00 and GVA grant CIAICO/ 2021/159. J.B. is a recipient of a "Práctica de Empresa - Plan de ayudas ai2 a la I+D+i 2022". A.P. is a recipient of a "Práctica de Empresa - Plan de ayudas ai2 a la I+D+i 2021". H.D.I. holds an "Contrato predoctoral para la formación de doctores, convocatoria 2021" (PRE2021-098767) from the Agencia Estatal de Investigación. A.A.G. thanks Grant PAID-01-21 Programa de Ayudas de Investigación y Desarrollo - Universitat Politècnica de València. Y.B. thanks Grant PAID-10-21 Acceso al Sistema Español de Ciencia e Innovación-Universitat Politècnica de València. Y.B. also thanks to Secretaría de Educación Superior, Ciencia, Tecnología e Innovación of Ecuador (Scholarship Convocatoria Abierta 2011).

References

Beal, J., Farny, N. G., Haddock-Angelli, T., Selvarajah, V., Baldwin, G. S., Buckley-Taylor, R., Gershater, M., Kiga, D., Marken, J., Sanchania, V., et al., 2020. Robust estimation of bacterial cell count from optical density. Communications biology 3 (1), 512.

Beal, J., Telmer, C. A., Vignoni, A., Boada, Y., Baldwin, G. S., Hallett, L., Lee, T., Selvarajah, V., Billerbeck, S., Brown, B., et al., 2022. Multicolor plate reader fluorescence calibration. Synthetic Biology 7 (1), ysac010.

Boada, Y., Picó, J., Vignoni, A., 2021. Multi-objective optimization tuning framework for kinetic parameter selection and estimation. Methods in Molecular Biology, vol 2385. Springer US, New York, NY.
DOI: 10.1007/978-1-0716-1767-0

Boada, Y., Santos-Navarro, F., Vignoni, A., Picó, J., 2022a. Optimization of the dynamic regulation in a branch-in metabolic pathway. IFAC-PapersOnLine 55 (7), 119–124.

Boada, Y., Santos-Navarro, F. N., Picó, J., Vignoni, A., 2022b. Modeling and optimization of a molecular biocontroller for the regulation of complex metabolic pathways. Frontiers in Molecular Biosciences 9.

Boada, Y., Vignoni, A., Alarcon-Ruiz, I., Andreu-Vilarroig, C., Monfort-Llorens, R., Requena, A., Picó, J., 2019a. Characterization of Gene Circuit Parts Based on Multiobjective Optimization by Using Standard Calibrated Measurements. ChemBioChem 20 (20).
DOI: 10.1002/cbic.201900272

Boada, Y., Vignoni, A., Picó, J., 2019b. Multiobjective identification of a feedback synthetic gene circuit. IEEE Transactions on Control Systems Technology 28 (1), 208–223.

Bryant Jr, J. A., Kellinger, M., Longmire, C., Miller, R., Wright, R. C., 2023. Assemblytron: Flexible automation of dna assembly with opentrons ot-2 lab robots. Synthetic Biology 8 (1), ysac032.

Buecherl, L., Myers, C. J., 2022. Engineering genetic circuits: advancements in genetic design automation tools and standards for synthetic biology. Current opinion in microbiology 68, 102155.

Carbonell, P., Le Feuvre, R., Takano, E., Scrutton, N. S., 2020. In silico design and automated learning to boost next-generation smart biomanufacturing. Synthetic Biology 5 (1), ysaa020.

Cummins, B., Vrana, J., Moseley, R. C., Eramian, H., Deckard, A., Fontanarrosa, P., Bryce, D., Weston, M., Zheng, G., Nowak, J., et al., 2023. Robustness and reproducibility of simple and complex synthetic logic circuit designs using a dbtl loop. Synthetic Biology 8 (1), ysad005.

González-Cebrián, A., Borràs-Ferrís, J., Boada, Y., Vignoni, A., Ferrer, A., Picó, J., 2023. Platero: A calibration protocol for plate reader green fluorescence measurements. Frontiers in Bioengineering and Biotechnology 11.

Gurdo, N., Volke, D. C., McCloskey, D., Nikel, P. I., 2023. Automating the design-build-test-learn cycle towards next-generation bacterial cell factories. New Biotechnology 74, 1–15.
URL: https://www.sciencedirect.com/science/article/pii/S187167842300002X
DOI: https://doi.org/10.1016/j.nbt.2023.01.002

Kang, D. H., Ko, S. C., Heo, Y. B., Lee, H. J., Woo, H. M., 2022. Robomoclo: A robotics-assisted modular cloning framework for multiple gene assembly in biofoundry. ACS Synthetic Biology 11 (3), 1336–1348.

Ko, S. C., Cho, M., Lee, H. J., Woo, H. M., 2022. Biofoundry palette: Planning-assistant software for liquid handler-based experimentation and operation in the biofoundry workflow. ACS Synthetic Biology 11 (10), 3538–3543.

MathWorks, 2023. How the genetic algorithm works.
URL: https://es.mathworks.com/help/gads/how-the-genetic-algorithm-works.html

Morgan, K., 2023. Plasmids 101: Origin of replication.
URL: https://blog.addgene.org/plasmid-101-origin-of-replication

Radivojević, T., Costello, Z., Workman, K., Garcia Martin, H., 2020. A machine learning automated recommendation tool for synthetic biology. Nature communications 11 (1), 4879.

Tellechea-Luzardo, J., Otero-Muras, I., Goñi-Moreno, A., Carbonell, P., 2022. Fast biofoundries: coping with the challenges of biomanufacturing. Trends in biotechnology.

Vidal, G., Vidal-Céspedes, C., Muñoz Silva, M., Castillo-Passi, C., Yáñez Feliú, G., Federici, F., Rudge, T. J., 2022a. Accurate characterization of dynamic microbial gene expression and growth rate profiles. Synthetic Biology 7 (1), ysac020.

Vidal, G., Vidal-Céspedes, C., Rudge, T. J., 2022b. Loica: Integrating models with data for genetic network design automation. ACS Synthetic Biology 11 (5), 1984–1990.

Vidal-Peña, G., 2023. Protocol unified design unit.
URL: https://github.com/RudgeLab/PUDU

Yanez Feliu, G., Earle Gomez, B., Codoceo Berrocal, V., Munoz Silva, M., Nunez, I. N., Matute, T. F., Arce Medina, A., Vidal, G., Vidal Cespedes, C., Dahlin, J., et al., 2020. Flapjack: Data management and analysis for genetic circuit characterization. ACS Synthetic Biology 10 (1), 183–191.

Appendix B

MATLAB scripts

Estimation of cell growth rate (μ).

```
function [estimated_mu] = Mu_estimate(Input_time_series,sampling_time)
%Estimates the specific growth rate
% We obtain the derivative as
% dy(k)/dt = (-y(k+2) + 8*y(k+1)-8*y(k-1) + y(k-2))/12/dt
% and then apply a Butterworth filter:
% [b,a] = butter(2,4/33);
% data_ff = filtfilt(b,a,data);
% Input_time_series is the sampled data of absorbance or Particles
% Sampling is assumed to be in minutes
% estimated_mu gives the estimate of the specific growth rate in hours^{-1}
%
% We first extend the vector on the left and right extremes:
x0_value = Input_time_series(1);
xf_value= Input_time_series(end);
temp_data= [x0_value;x0_value;Input_time_series;xf_value;xf_value];
der_temp_data = (-temp_data(5:end,:)+8*temp_data(4:end-1,:)-8*temp_data(2:end-3,:)+ ...
    temp_data(1:end-4,:))./(12*sampling_time);
raw_estimated_mu = der_temp_data./Input_time_series;
%[b,a] = butter(2,8/33); %wn=4/33
b=[0.0928, 0.1857, 0.0928];
a=[1, -0.9728, 0.3441];
mu=filtfilt(b,a,raw_estimated_mu);
if mu<=0
   estimated_mu=1e-4;
else
   estimated_mu=mu;
end
end
```

Function to analyze experimental results.

```
load pLux_pASSO14.mat;
startpoint=3;
ControlStrain=[]; %1 IF ANY
user_end_time =[8];
Font_size=12;

%SETTNGS CONTROL STRAIN
strain=char(evol.medios.strains.Id);
stepTime = evol.time_OD(2)-evol.time_OD(1); %every X minutos
timePRE = evol.npre*stepTime/60-startpoint*stepTime/60;
timeEXP = (evol.time_OD(startpoint:end)-evol.time_OD(startpoint))./60; %in hours
if isempty(user_end_time)
    tEnd=timeEXP(end);
    fpoint = length(timeEXP);
else
    tEnd =user_end_time;
    [fpoint,v]=find(timeEXP>=tEnd & timeEXP<=tEnd+stepTime/60);
    timeEXP=timeEXP(startpoint:fpoint)-timeEXP(startpoint);
end

 %Control strain to correct background
if isempty(ControlStrain)
   ControlSt=0; Rcontrol=zeros(length(timeEXP),1);
else
   ControlSt=size(strain,1);
   CSt_AHL=char(evol.medios.strains(ControlSt).data.Inducc);
   if CSt_AHL > 1
       ODcontrol = evol.medios.strains(ControlSt).data(size(CSt_AHL,1)).OD_media(startpoint:end,:)- ...
       evol.medios.OD_medio(startpoint:end);
       Fcontrol = evol.medios.strains(ControlSt).data(size(CSt_AHL,1)).F_media(startpoint:end,:)- ...
       evol.medios.F_medio(startpoint:end);
       [Rcontrol, Rlog, Particles, MEF] = Calibrate_MEFL_GFPmut3b(ODcontrol,Fcontrol);
   end
end

mypal = ['00adff', '4fbee4', '58cfe4', '5cdfea', '5deff3', '5cffff', 'ede12a', 'eec14b', 'f19d6a',
     'f2738d', 'f033b4'];
map = hex2rgb(mypal);
```

```
%MEAN AND TEMPORAL FIGURE
%Main programm
strain=char(evol.medios.strains.Id);

stepTime = evol.time_OD(2)-evol.time_OD(1); %every X minutos
timePRE = evol.npre*stepTime/60-startpoint*stepTime/60;
timeEXP = (evol.time_OD(startpoint:end)-evol.time_OD(startpoint))./60; %in hours
if isempty(user_end_time)
    tEnd=timeEXP(end);
else
    tEnd =user_end_time;
end

 %Control strain to correct background
if isempty(ControlStrain)
   ControlSt=0; Rcontrol=zeros(length(timeEXP),1);
else
   ControlSt=size(strain,1);
   CSt_AHL=char(evol.medios.strains(ControlSt).data.Inducc);
   if CSt_AHL > 1
       ODcontrol = evol.medios.strains(ControlSt).data(size(CSt_AHL,1)).OD_media(startpoint:end,:)- ...
       evol.medios.OD_medio(startpoint:end);
       Fcontrol = evol.medios.strains(ControlSt).data(size(CSt_AHL,1)).F_media(startpoint:end,:)- ...
       evol.medios.F_medio(startpoint:end);
       [Rcontrol, Rlog, Particles, MEF] = Calibrate_MEFL_GFPmut3b(ODcontrol,Fcontrol);
   end
end

mypal = ['00adff', '4fbee4', '58cfe4', '5cdfea', '5deff3', '5cffff', 'ede12a', 'eec14b', 'f19d6a',
     'f2738d', 'f033b4'];
map = hex2rgb(mypal);
```

```
for m=1:size(strain,1)
    figure('Name',strain(m,:));
    AHL=char(evol.medios.strains(m).data.Inducc);
    total_ax=3;
    for j=1:size(AHL,1)
            ODblank = evol.medios.strains(m).data(j).OD_media(startpoint:end,:)- ...
            evol.medios.OD_medio(startpoint:end);
            Fblank = evol.medios.strains(m).data(j).F_media(startpoint:end,:)- ...
            evol.medios.F_medio(startpoint:end);
            [R, Rlog, Particles, MEF] = Calibrate_MEFL_GFPmut3b(ODblank,Fblank);
            for n=1:total_ax
                hs=subplot(total_ax,1,n,'xlim',[0 tEnd]);
                if n==1
                    plot(timeEXP, ODblank,'Color',map(j,:),'LineWidth',1);
                    hold on;
                    if j==size(AHL,1)
                        xlabel('Time (h)');
                        ylabel(' OD_{600}'); title(strain(m,:));
                        grid on;
                        pos=get(hs, 'position');
                        annotation("line",(pos(1)+timePRE/timeEXP(end)*(pos(3))).*[1 1],...
                        [pos(2) pos(2)+pos(4)],'LineWidth',1,'LineStyle',"--");
                    end

                elseif n==2
                    plot(timeEXP, MEF,'Color',map(j,:),'LineWidth',1)
                    hold on;
                    if j==size(AHL,1)
                        lgd=legend(AHL,'Position',[0.874 0.363 0.112 0.29]);
                        title(lgd,'AHL (nM)');
                        xlabel('Time (h)');
                        ylabel('MEFL');
                        grid on;
                        pos=get(hs, 'position');
                        annotation("line",(pos(1)+timePRE/timeEXP(end)*(pos(3))).*[1 1],...
                        [pos(2) pos(2)+pos(4)],'LineWidth',1,'LineStyle',"--");
                    end

                  else
                      if m==ControlSt
                          Rcorrected=R;
                      else
                          Rcorrected=R-Rcontrol;
                      end
                      semilogy(timeEXP,Rcorrected,'Color',map(j,:),'LineWidth',1)
                      hold on;
                      if j==size(AHL,1)
                           xlabel('Time (h)');
                           ylabel('FOD (MEFL/Particles)');
                           grid on;
                           pos=get(hs, 'position');
                           annotation("line",(pos(1)+timePRE/timeEXP(end)*(pos(3))).*[1 1],...
                               [pos(2) pos(2)+pos(4)],'LineWidth',1,'LineStyle',"--");
                      end
                  end
            end
    end
end
```

```
%HILL FUNCTION
t1=5; t2=6; %Times for Hill Function
setime = round(t1*60/5);
endtime = round(t2*60/5);
hf=figure('Name','Hill function');

for m=1:(size(strain,1)-1)
    AHL=char(evol.medios.strains(m).data.Inducc);

    if size(AHL,1)>1
        expression=zeros(1,size(AHL,1));
        R_std=zeros(1,size(AHL,1));

        inductions=str2num(AHL(:,:));
        inductions(1)=0.01;
        for j=1:size(AHL,1)
            ODblank=evol.medios.strains(m).data(j).ODblank(setime:endtime,:);
            PARTICLES=evol.medios.strains(m).data(j).Particles(setime:endtime,:);
            MEF=evol.medios.strains(m).data(j).MEFL(setime:endtime,:);
            R=evol.medios.strains(m).data(j).LogMEFcell(setime:endtime,:); %Logaritmit FOD
            expression(1,j) = mean(mean(R,2),1);
            R_var = mean(var(R,0,1)) + var(mean(R,1)); %law of total variance
            R_std(1,j) = sqrt(R_var);
            if m==1
                hp1=errorbar(inductions(j),expression(1,j),R_std(1,j),'o','MarkerSize',5,...
                    'MarkerFaceColor',map(j,:),'Color',map(j,:),'MarkerEdgeColor','k','LineWidth',1);
            elseif m==2
                hp2=errorbar(inductions(j),expression(1,j),R_std(1,j),'s','MarkerSize',5,...
                    'MarkerFaceColor',map(j,:),'Color',map(j,:),'MarkerEdgeColor','k','LineWidth',1);
            else
                hp3=errorbar(inductions(j),expression(1,j),R_std(1,j),'v','MarkerSize',5,...
                    'MarkerFaceColor',map(j,:),'Color',map(j,:),'MarkerEdgeColor','k','LineWidth',1);
            end
            hold on;
        end
        loglog(inductions,expression,'--k');
    else
        close(gcf);
    end
end
colormap(hf,map);
colorbar('southoutside','TickLabels',cellstr(AHL));
xlabel('AHL (nM)'); ylabel({'log(FOD)','(MEFL/Particles)'});
title('Hill functions')
if m==3
    lgd=legend([hp1,hp2,hp3],strain(:,:),'Location','northwest');
else
    lgd=legend([hp1,hp2],strain(:,:),'Location','northwest');
end
title(lgd,'Strain');
grid on;
```

```
%NORMALIZED HILL FUNCTION
total_ax=2;

for m=1:(size(strain,1)-1)
    figure('Name',['Hill-like' strain(m,:)]);
    AHL=char(evol.medios.strains(m).data.Inducc);

    if size(AHL,1)>1
       expression=zeros(1,size(AHL,1));
       R_std=zeros(1,size(AHL,1));

       for j=1:size(AHL,1)
            ODblank=evol.medios.strains(m).data(j).ODblank(setime:endtime,:);
            PARTICLES=evol.medios.strains(m).data(j).Particles(setime:endtime,:);
            MEF=evol.medios.strains(m).data(j).MEFL(setime:endtime,:);
            R=evol.medios.strains(m).data(j).MEFLcell(setime:endtime,:);
            expression(1,j) = mean(mean(R,2),1);
            R_var = mean(var(R,0,1)) + var(mean(R,1)); %law of total variance
            R_std(1,j) = sqrt(R_var);
       end
       Maxvalue=max(expression);
       Nexpression=expression./Maxvalue;

       hs=subplot(1,2,1);
       for k=1:size(expression,2)
            errorbar(inductions(k),expression(1,k),R_std(1,k),'d','MarkerSize',6,'MarkerFaceColor', ...
            map(k,:),'Color',map(k,:),'MarkerEdgeColor','k','LineWidth',1);
            hold on;
       end
            plot(hs,inductions,expression,'--k');
            xlabel('AHL (nM)','FontSize',14); ylabel({'FOD','(MEFL/Particles)'},'FontSize',14);
            title(strain(m,:),'FontSize',16);
            lgd=legend(AHL,'Location','southeast');
            title(lgd,'AHL (nM)');
            grid on;

        hs=subplot(1,2,2);
        for k=1:size(expression,2)
            aux=num2str(inductions);
            X = categorical(cellstr(aux));
            X = reordercats(X,cellstr(aux)); %ensure the re-ordering
            hb=bar(X(k),Nexpression(1,k),'FaceColor', map(k,:));
            hold on;
        end
        title(strain(m,:),'FontSize',16);
        xlabel('AHL (nM)','FontSize',14); ylabel('Promoter Activity (adim)','FontSize',14)
        grid on;
    else
        close(gcf);
    end
end
```

Definition of cost function.

```
function J=cost_function_pLux(params)
% Cost function to estimate parameters of Model for pLux Promoter
% Model: Contant PARAMETERS with Mu_max
% Parameters:
% kg pLux Promoter Strength
% kdlux pLux promoter dissosiation constant
% alpha promoter basal expression
% pG RBS Strength: depends on condition (5)
% CN: Copy number: depends on condition (2), We fix the low copy to 5 and
% the high copy will be unknown relative to that.
% dmG: mRNA degradation rate same for all the conditions
% dG: Protein degradation rate same for all the conditions
% dA: AHL degradation
% Mu_max -> From experimental data
%
% params = [kR kdlux alpha pG1 pG2 pG3 pG4 pG5 CNh dmG dG dA]
%
% Updated 10/06/2023 by AV and Yadira Boada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% General parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Experimental data
load Data_struct.mat
J_parcial = 0;
J_RSE_parcial = 0;
dibujar = 1;
% pLux Promoter
p = parameters();
p.kg = params(1);
p.kdlux =params(2);
p.alpha = params(3);
% RBS
pg = params(4:8);
% Copy number
CNh = params(9);
CN = [5*CNh 5];
% mRNA degradation
p.dmg = params(10);
p.dmR = p.dmg;
% Protein degradation
p.dg = params(11);
p.dR = p.dg;
% AHL degradation
p.dA = params(12); % 0.00004
p.dAe = p.dA; % 0.0000481

% Loop to cover all the RBS
for ii = 1:5
    p.pg = pg(ii);
```

```
%Experiment details
strain=char(Data{ii}.medios.strains.Id);
% loop for Copy Numbers (high and low)
for m=1:2 % strain 1 is HIgh copy, 2 is low copy
    p.CN = CN(m);
    AHL = char(Data{ii}.medios.strains(m).data.Inducc);
    Label_induction={};

    %Experimental inductions
    induction = str2num(AHL(:,:)); %Induction in the lab [nM]
    %% Main
    %Colormap
    yel_purple12=['ffb400','db9e17','b98725','99712f','795c38','463831','443f6c','585092', ...
    '665bb3','7b6fce','9b90eb','ccc5ff'];
    map = hex2rgb(yel_purple12);

    %Create tables for saving data
    Tmolec=table;
    Tconc=table;

    startpoint=1;
    stepTime = Data{ii}.time_OD(2)-Data{ii}.time_OD(1); %every X minutos
    DurationPRE= Data{ii}.npre*stepTime/60-startpoint*stepTime/60;
    timeEXP=(Data{ii}.time_OD(startpoint:end)-Data{ii}.time_OD(startpoint)); %minutes
    timeINDUC = timeEXP(Data{ii}.npre+1)-timeEXP(Data{ii}.npre)-stepTime;
    if timeINDUC>stepTime
        timeEXP=timeEXP-[zeros(Data{ii}.npre,1); timeINDUC.*ones(length(timeEXP)-Data{ii}.npre,1)];
    end
    t2=6;
    time_hill = round(t2*60/5);

    %Lab volumes %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Vext_0 = 3e-3; %mL
    Vext_plate = 200e-6; %uL

    %ODE
    options = odeset('AbsTol',1e-8,'RelTol',1e-6); % for ode function

    %Average growth rate
    mu_matrix=[];
    for j=1:size(AHL,1)
        mu_matrix(:,j) = Data{ii}.medios.strains(m).data(j).mu; %HIGH/Low copy
    end
    muEXP= mean(mu_matrix(:,1:8),2); %
    mu_null = max(muEXP);

    gfp_matrix=[];
    od_matrix=[];
    for j=1:size(AHL,1)
        od_matrix(:,j) = mean(Data{ii}.medios.strains(m).data(j).Particles(2,:));
        gfp_matrix(:,j) = mean(Data{ii}.medios.strains(m).data(j).MEFL(2,:));
        od_matrix_i(:,j) = mean(Data{ii}.medios.strains(m).data(j).Particles(Data{ii}.npre,:));
        gfp_matrix_i(:,j) = mean(Data{ii}.medios.strains(m).data(j).MEFL(Data{ii}.npre,:));
    end
    gfp_initial= mean(gfp_matrix)/mean(od_matrix);
    gfp_initial_i= mean(gfp_matrix_i)/mean(od_matrix_i);
    %%Simulation
    Variance = 0; Stdeviation =0; Ncell = 1;
```

```
%System size
promoters={'Plux'};
phi={'Phi'};
species={'LuxR','GFP','AHL','OD600','AHLe'};
[~,ODindex] = find(strcmp('OD600',species)==1);
[~,AHLindex] = find(strcmp('AHLe',species)==1);
[~,GFPindex] = find(strcmp('GFP',species)==1);
States = length(species);
for j=1:size(AHL,1)
    Label_induction{j}=deblank(strrep(AHL(j,:), '.', '_'));
    ODblank = Data{ii}.medios.strains(m).data(j).OD_media(startpoint:end,:)- ...
    Data{ii}.medios.OD_medio(startpoint:end);
    ODinitial = ODblank(1);
    ODmax = ODblank(end);
    %muEXP= Data{ii}.medios.strains(m).data(j).mu;

    %% 0 Null initial conditions %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    p.mu = mu_null; Cellinitial = 1;
    Initial = [zeros(1,States-2) Cellinitial 0]; %ini conditions[species, cells, ahle]
    tfin = 60*16; %simulation time
    tspan = 0:tfin-stepTime;
    Texp = tspan;
    Gexp = p.mu.*ones(length(Texp),1);
    [t0,x0] = ode23t(@(t,x) model_exp_pASSO1_mu_max(t,x,Texp,Gexp,p),tspan, Initial, options);

    %% 1 PRE-INDUCTION & GROWTH %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Cellinitial = ODinitial*p.Vext*p.OD_to_cells;
    %Initial = x0(end,:)
    Initial = [219089 gfp_initial x0(end,3) Cellinitial 0]; %Initial conditions
    tfin = 60*DurationPRE; % Tiempo de simulacion (min)
    tspan = timeEXP(1:Data{ii}.npre);
    Texp = tspan;
    Gexp = muEXP(1:length(tspan));
    [t1,x1] = ode23t(@(t,x) model_exp_pASSO1_mu_max(t,x,Texp,Gexp,p),tspan, Initial, options);

    %% 2 Adding ahle %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    ahle0 = induction(j).*p.ahl_to_molec; %AHLe vector
    tfin = timeEXP(end); % Tiempo de simulacion (min)
    tspan = timeEXP(Data{ii}.npre+1):stepTime:tfin;
    tspan = timeEXP(Data{ii}.npre):stepTime:tfin;
    Texp = timeEXP(length(t1)+1:length(timeEXP));
    Gexp = muEXP(length(t1)+1:length(timeEXP));
    Initial = [x1(end,1) gfp_initial_i x1(end,3:4) ahle0];
    %Initial = [x1(end,1:end-1) ahle0]; %Initial conditions
    [t2,x2] = ode23t(@(t,x) model_exp_pASSO1_mu_max(t,x,Texp,Gexp,p),tspan, Initial, options);
    for k=1:length(species)
        if k==ODindex
            data = [x1(:,k); x2(:,k)]./(p.OD_to_cells*p.Vext);
            data2= data;
        elseif k==length(species)
            data = [x1(:,k); x2(:,k)];
            data2 = [x1(:,k); x2(:,k)].*p.ahlmolec_to_nM;
        else
            data = [x1(:,k); x2(:,k)];
            data2 = data.*p.molec_to_nM;
        end
        Tmolec(:,k) =table(data);
        Tconc(:,k) =table(data2);
    end
```

```
            Tmolec.Properties.VariableNames = species;
            Tconc.Properties.VariableNames = species;
            field=['AHL_' Label_induction{j}];
            out.(field).molec=table2struct(Tmolec,"ToScalar",true);
            out.(field).conc=table2struct(Tconc,"ToScalar",true);
            out.time=timeEXP./60;
            % Experimental data for the ijk construction
            part= mean(Data{ii}.medios.strains(m).data(j).Particles,2);
            gfp= mean(Data{ii}.medios.strains(m).data(j).MEFL,2);
            measured = gfp./part;
            % Simulated data for the ijk construction
            simulated = Tmolec.("GFP"); %del modelo constante, si quisieramos el modelo con Psi es
                Data_model_molec{i,j,k}.("GFP mu exp")
            is11 = (m==1) * (ii==1);
            %J_parcial = J_parcial + sum((measured(Data{ii}.npre:end)-simulated(Data{ii}.npre:end)).^2)
                ./length(measured(Data{ii}.npre:end));
            J_RSE_parcial = J_RSE_parcial + ~is11 * sum(
                ((measured(Data{ii}.npre:end)-simulated(Data{ii}.npre:end))./(measured(Data{ii}.npre:end))
                ) .^2 ) ;


        end %For Induccion

        if dibujar == 1
            figure('Name',['Comparison ' strain(m,:)]);
            for k=1:length(induction)
                field=['AHL_' Label_induction{k}];
                field2=char(species(GFPindex));
                semilogy(out.time,log10(out.(field).molec.(field2)),'LineWidth',1.5,'Color',map(k,:));
                    %molecules or concentration
                hold on;
                Particles_avr=mean(Data{ii}.medios.strains(m).data(k).Particles,2);
                MEFL_avr=mean(Data{ii}.medios.strains(m).data(k).MEFL,2);
                semilogy(out.time,log10(MEFL_avr./Particles_avr),':','LineWidth',1.5,'Color',map(k,:));
            end
            grid on;
            xlabel('time (hours)'); ylabel('GFP (molec)');
        end
    end % For copy number
end % For RBS
%J = J_parcial/(ii*m*j);
J = J_RSE_parcial/(ii*m*j);
end
```

Bibliography

Benner, S. A., & Sismour, A. M. (2005). Synthetic biology [Number: 7 Publisher: Nature Publishing Group]. *Nature Reviews Genetics*, *6*(7), 533–543. https://doi.org/10.1038/nrg1637

Boada, Y., Vignoni, A., Alarcon-Ruiz, I., Andreu-Vilarroig, C., Monfort-Llorens, R., Requena, A., & Picó, J. (2019). Characterization of gene circuit parts based on multiobjective optimization by using standard calibrated measurements. *ChemBioChem*, *20*(20), 2653–2665. https://doi.org/10.1002/cbic.201900272

Boada Acosta, Y. F. (2018, October 10). *A systems engineering approach to model, tune and test synthetic gene circuits* (Doctoral dissertation). Universitat Politècnica de València. Valencia (Spain). https://doi.org/10.4995/Thesis/10251/112725

Brown, J. (2007). The iGEM competition: Building with biology. *IET Synthetic Biology*, *1*(1), 3–6. https://doi.org/10.1049/iet-stb:20079020

BsaI-HF®v2. (n.d.). https://international.neb.com/products/r3733-bsai-hf-v2#Product%20Information

BsmBI. (n.d.). https://international.neb.com/products/r0134-bsmi#Product%20Information

Buchler, N. E., Gerland, U., & Hwa, T. (2005). Nonlinear protein degradation and the function of genetic circuits. *Proceedings of the National Academy of Sciences*, *102*(27), 9559–9564. https://doi.org/10.1073/pnas.0409553102

Carson, N. (2020, May 29). *English: 96-well plate*. Retrieved June 21, 2023, from https://commons.wikimedia.org/wiki/File:96-Well_plate.svg

Cheng, A. A., & Lu, T. K. (2012). Synthetic biology: An emerging engineering discipline [_eprint: https://doi.org/10.1146/annurev-bioeng-071811-150118]. *Annual Review of Biomedical Engineering*, *14*(1), 155–178. https://doi.org/10.1146/annurev-bioeng-071811-150118

Clark, J. (2013, October 3). *The beer-lambert law* [Chemistry LibreTexts]. Retrieved June 21, 2023, from https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Spectroscopy/Electronic_Spectroscopy/Electronic_Spectroscopy_Basics/The_Beer-Lambert_Law

Cloud-based platform for biotech r&d | benchling. (n.d.). Retrieved June 17, 2023, from https://www.benchling.com/

Crick, F. (1970). Central dogma of molecular biology. *Nature*, *227*(5258), 561–563. https://doi.org/10.1038/227561a0

Curriculum modules | EBRC. (n.d.). Retrieved June 21, 2023, from https://ebrc.org/curriculum-modules/

DBTL cycle - design. (2021). Retrieved May 16, 2023, from https://2021.igem.org/Engineering/Design

Ecuador iGEM team. (2021). Retrieved May 27, 2023, from https://2021.igem.org/Team:Ecuador/Engineering

Érdi, P., & Tóth, J. (1989). *Mathematical models of chemical reactions: Theory and applications of deterministic and stochastic models* [Google-Books-ID: iDu8AAAAIAAJ]. Manchester University Press.

Fluorescence intensity measurements | BMG LABTECH. (n.d.). Retrieved June 25, 2023, from https://www.bmglabtech.com/en/fluorescence-intensity/

Fraile López, A. (2020, September 8). *Diseño y aplicabilidad de un circuito genético basado en el control antitético en la síntesis de proteínas y metabolitos* (Proyecto/Trabajo fin de carrera/grado) [Accepted: 2020-09-08T09:08:40Z]. Universitat Politècnica de València. Retrieved June 16, 2023, from https://riunet.upv.es/handle/10251/149571

Gesztelyi, R., Zsuga, J., Kemeny-Beke, A., Varga, B., Juhasz, B., & Tosaki, A. (2012). The hill equation and the origin of quantitative pharmacology. *Archive for History of Exact Sciences*, *66*(4), 427–438. https://doi.org/10.1007/s00407-012-0098-5

Goutelle, S., Maurin, M., Rougier, F., Barbaut, X., Bourguignon, L., Ducher, M., & Maire, P. (2008). The hill equation: A review of its capabilities in pharmacological modelling [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1472-8206.2008.00633.x]. *Fundamental & Clinical Pharmacology*, *22*(6), 633–648. https://doi.org/10.1111/j.1472-8206.2008.00633.x

Hill, A. V. (1910). The possible effects of the aggregation of the molecules of hæmoglobin on its dissociation curves. *The Journal of Physiology*.

Hirst, C. D. (2014). Automated BioPart characterisation for synthetic biology. *Imperial College London*.

Hobom, B. (1980). Surgery of genes-at the doorstep of synthetic biology. *75*(24), 14–21.

How the genetic algorithm works - MATLAB & simulink. (n.d.). Retrieved June 25, 2023, from https://es.mathworks.com/help/gads/how-the-genetic-algorithm-works.html

Johansson, R. (1993). *System modeling and identification*. Prentice Hall.

Karplus, W. J. (1977). The spectrum of mathematical modeling and systems simulation. *Mathematics and Computers in Simulation*, *19*(1), 3–10. https://doi.org/10.1016/0378-4754(77)90034-9

Kaufmann, G. F., Sartorio, R., Lee, S.-H., Rogers, C. J., Meijler, M. M., Moss, J. A., Clapham, B., Brogan, A. P., Dickerson, T. J., & Janda, K. D. (2005). Revisiting quorum sensing: Discovery of additional chemical and biological functions for 3-oxo-n-acylhomoserine lactones. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(2), 309–314. https://doi.org/10.1073/pnas.0408639102

Kriz, A., Schmid, K., Baumgartner, N., Ziegler, U., Berger, I., Ballmer-Hofer, K., & Berger, P. (2010). A plasmid-based multigene expression system for mammalian cells [Number: 1 Publisher: Nature Publishing Group]. *Nature Communications*, *1*(1), 120. https://doi.org/10.1038/ncomms1120

Kubitschek, H. E., & Friske, J. A. (1986). Determination of bacterial cell volume with the coulter counter. *Journal of Bacteriology*, *168*(3), 1466–1467. https://doi.org/10.1128/jb.168.3.1466-1467.1986

Lade, H., Paul, D., & Kweon, J. H. (2014). Quorum quenching mediated approaches for control of membrane biofouling. *International Journal of Biological Sciences*, *10*(5), 550–565. https://doi.org/10.7150/ijbs.9028

Lambert, T. (n.d.). *GFPmut3 at FPbase* [FPbase]. Retrieved June 25, 2023, from https://www.fpbase.org/protein/gfpmut3/

Lipkin, L., & Smith, D. (2005). Logistic growth model. Retrieved July 3, 2023, from https://amser.org/r13768/logistic_growth_model

M, P. (2021, October 28). *End-to-end introduction to evaluating regression models* [Analytics vidhya]. Retrieved June 18, 2023, from https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/

Real Decreto 1098/2001, de 12 de octubre, por el que se aprueba el Reglamento general de la Ley de Contratos de las Administraciones Públicas (2001, October 26). Retrieved July 4, 2023, from https://www.boe.es/eli/es/rd/2001/10/12/1098

Najafpour, G. D. (2007, January 1). CHAPTER 5 - growth kinetics. In G. D. Najafpour (Ed.), *Biochemical engineering and biotechnology* (pp. 81–141). Elsevier. https://doi.org/10.1016/B978-044452845-2/50005-7

Novák, B., & Tyson, J. J. (2004). A model for restriction point control of the mammalian cell cycle. *Journal of Theoretical Biology*, *230*(4), 563–579. https://doi.org/10.1016/j.jtbi.2004.04.039

OT-2 liquid handler | opentrons lab automation | opentrons. (n.d.). Retrieved June 26, 2023, from https://opentrons.com/products/robots/ot-2/

Pai, A., & You, L. (2009). Optimal tuning of bacterial sensing potential. *Molecular Systems Biology*, *5*, 286. https://doi.org/10.1038/msb.2009.43

Picó Marco, J. A. (2021). *Biología computacional*.

Requena Gutiérrez, A. (2018a). Part:BBa_k2656003. http://parts.igem.org/Part:BBa_K2656003

Requena Gutiérrez, A. (2018b). Part:BBa_k2656016. http://parts.igem.org/Part:BBa_K2656016

Requena Gutiérrez, A. (2018c). Part:BBa_k2656026. http://parts.igem.org/Part:BBa_K2656026

Requena Gutiérrez, A. (2019, September 3). *Diseño y optimización de un circuito genético para el control interno de la densidad poblacional de un cultivo de escherichia coli* (Proyecto/-Trabajo fin de carrera/grado) [Accepted: 2019-09-03T09:43:33Z]. Universitat Politècnica de València. Retrieved June 11, 2023, from https://riunet.upv.es/handle/10251/124840

Requena Gutiérrez, A., & Ropero, C. (2018a). Part:BBa_k2656004. http://parts.igem.org/Part:BBa_K2656004

Requena Gutiérrez, A., & Ropero, C. (2018b). Part:BBa_k2656008. http://parts.igem.org/Part:BBa_K2656008

Requena Gutiérrez, A., & Ropero, C. (2018c). Part:BBa_k2656009. http://parts.igem.org/Part:BBa_K2656009

Requena Gutiérrez, A., & Ropero, C. (2018d). Part:BBa_k2656010. http://parts.igem.org/Part:BBa_K2656010

Requena Gutiérrez, A., & Ropero, C. (2018e). Part:BBa_k2656011. http://parts.igem.org/Part:BBa_K2656011

Requena Gutiérrez, A., & Ropero, C. (2018f). Part:BBa_k2656012. http://parts.igem.org/Part:BBa_K2656012

Requena Gutiérrez, A., & Ropero, C. (2018g). Part:BBa_k2656022. http://parts.igem.org/Part:BBa_K2656022

Sarrion-Perdigones, A., Falconi, E. E., Zandalinas, S. I., Juárez, P., Fernández-del-Carmen, A., Granell, A., & Orzaez, D. (2011). GoldenBraid: An iterative cloning system for standardized assembly of reusable genetic modules [Publisher: Public Library of Science]. *PLOS ONE*, *6*(7), e21622. https://doi.org/10.1371/journal.pone.0021622

Segel, L. A., & Slemrod, M. (1989). The quasi-steady-state assumption: A case study in perturbation. *SIAM Review*, *31*(3), 446–477. https://doi.org/10.1137/1031091

Sun, S., Wheeler, M., Obeyesekere, M., & Patrickjr, C. (2005). A deterministic model of growth factor-induced angiogenesis. *Bulletin of Mathematical Biology*, *67*(2), 313–337. https://doi.org/10.1016/j.bulm.2004.07.004

Sutton, S. (2006). *Measurement of cell concentration in suspension by optical density*. Retrieved June 10, 2023, from http://microbiologynetwork.com/measurement-of-cell-concentration-in-suspension-by-optical-density.asp

Synthetic biology [Genome.gov]. (2022, September 14). Retrieved May 15, 2023, from https://www.genome.gov/about-genomics/policy-issues/Synthetic-Biology

Synthetic Biology and Biosystems Control Lab – SB2CL. (n.d.). Retrieved June 25, 2023, from http://sb2cl.ai2.upv.es/

Szybalski, W., Kim, S. C., Hasan, N., & Podhajska, A. J. (1991). Class-IIS restriction enzymes — a review. *Gene*, *100*, 13–26. https://doi.org/10.1016/0378-1119(91)90345-C

T4 DNA ligase. (n.d.). https://international.neb.com/products/m0202-t4-dna-ligase#Product%20Information

TECHNÉ 3prime thermal cycler. (n.d.). Retrieved June 26, 2023, from https://jeulin.com/jeulin_en/591071.html

Trabelsi, H., Koch, M., & Faulon, J.-L. (2018). Building a minimal and generalizable model of transcription factor–based biosensors: Showcasing flavonoids [_eprint: https://onlinelibrary.wiley.com/doi/p *Biotechnology and Bioengineering*, *115*(9), 2292–2304. https://doi.org/10.1002/bit.26726

Xie, M., & Fussenegger, M. (2018). Designing cell function: Assembly of synthetic gene circuits for cell biology applications [Number: 8 Publisher: Nature Publishing Group]. *Nature Reviews Molecular Cell Biology*, *19*(8), 507–525. https://doi.org/10.1038/s41580-018-0024-z

Yang, X.-S. (2021, January 1). Chapter 6 - genetic algorithms. In X.-S. Yang (Ed.), *Nature-inspired optimization algorithms (second edition)* (pp. 91–100). Academic Press. https://doi.org/10.1016/B978-0-12-821986-7.00013-5

Zhang, G., Gurtu, V., & Kain, S. R. (1996). An enhanced green fluorescent protein allows sensitive detection of gene transfer in mammalian cells. *Biochemical and Biophysical Research Communications*, *227*(3), 707–711. https://doi.org/10.1006/bbrc.1996.1573