



Universidad
Zaragoza

1542

Trabajo Fin de Grado

Análisis y simulación de modelos metabólicos de *E. coli* para la producción de citramalato

*Analysis and simulation of metabolic models of
E. coli for citramalate production*

Autora

Violeta Gracia Román

Director

Jorge Emilio Júlvez Bueno

Universidad de Zaragoza
Escuela de Ingeniería y Arquitectura
2021

RESUMEN

La participación del ácido citramálico en el proceso de producción de termoplásticos ha provocado que la obtención de citramalato se convierta en una línea de estudio de gran interés. Los numerosos avances en biotecnología han permitido producir citramalato mediante la bacteria *Escherichia coli* modificada genéticamente, posibilitando que este proceso pueda ser aplicado a escala industrial a medio plazo.

Con el fin de disminuir el coste y el tiempo requerido por los experimentos de laboratorio, el área de la biología de sistemas está desarrollando modelos matemáticos que representan las reacciones químicas del metabolismo de la bacteria. A través de estos modelos matemáticos, es posible obtener los parámetros óptimos de operación que maximizan la producción de citramalato con la ayuda de métodos computacionales.

En este trabajo, se van a estudiar dos modelos matemáticos complementarios del metabolismo de la bacteria *Escherichia coli* modificada genéticamente: un modelo cinético y un modelo estequiométrico a escala genómica. El propósito del estudio es hallar los parámetros óptimos de operación para la producción de citramalato, considerando factores dinámicos de las reacciones y el mayor número posible de rutas metabólicas. Además, se analiza el flujo de alimentación de la bacteria con el fin de minimizar su valor, así como el número total de elementos presentes, sin que esto llegue a afectar en la obtención del ácido, con el objetivo de disminuir los costes totales del proceso. El estudio se ha llevado a cabo mediante la simulación y optimización de los modelos metabólicos en un entorno informático.

Índice

1. Introducción	1
1.1 Motivación.....	1
1.2 Objetivo.....	2
1.3 Metodología	5
1.3.1 El estado estacionario.....	5
1.3.2 Modelo cinético	6
1.3.3 Modelo estequiométrico.....	7
2. Análisis del modelo cinético.....	8
2.1 Adición de la reacción de citramalato.....	8
2.2 Variación de V_{max} para maximizar la productividad	9
2.3 Obtención de las cotas de flujo con más influencia en la síntesis del citramalato. ...	11
3. Modelo estequiométrico.....	13
3.1. Análisis del balance de flujos.....	13
3.2. Adición de la reacción de citramalato.....	14
3.3. Establecimiento de las cotas de flujo	15
3.3 Análisis de la influencia del flujo de citramalato en la productividad.	17
3.4. Minimización del medio de cultivo	19
3.5 Condiciones anaerobias.....	22
4. Resultados	23
4.1. Parámetros cinéticos.....	23
4.2. Flujos óptimos para el sistema	24
4.3. Minimización del medio.....	25
5. Planificación temporal y presupuesto	27
5.1 Planificación temporal del proyecto.....	27
5.2 Estimación de costes.....	28

6. Conclusiones	29
6.1 Conclusiones generales	29
6.2 Trabajos futuros	30
7. Bibliografía	31
Lista de Figuras	33
Lista de tablas	34
Anexo A	35
A.I Programa principal.....	35
A.II Optimización cinética.....	37
A.III Adición de la síntesis de citramalato al modelo estequiométrico	39
A.IV Análisis de balance de flujo	41

Capítulo 1

Introducción

1.1 Motivación

El ácido citramálico es un ácido hidrocarboxílico capaz de convertirse en ácido metacrílico (MAA) a través de procesos de descarboxilación y deshidratación. El MAA es un compuesto químico utilizado para la formación de polimetilmacrilato, un polímero de éster empleado como termoplástico transparente. Las principales aplicaciones de este tipo de plásticos se encuentran en los sectores de la construcción, el mobiliario, la iluminación y las tecnologías médicas [1].

Unos de los principales problemas a escala industrial, para la obtención de MMA, son los riesgos asociados a la manipulación y transporte de cianuros volátiles, necesarios para la obtención de este ácido. Este factor es uno de los principales motores para el estudio de la producción de MMA a través de procesos microbianos, desarrollados por ingeniería biológica [2]. La ingeniería metabólica y la biología sintética son sectores en crecimiento, que permiten la producción de productos bioquímicos mediante procesos sostenibles. La síntesis de sustancias químicas provenientes de los combustibles fósiles puede ser sustituida por el uso de microorganismos vivos, esta alternativa proporciona un recurso renovable de mayor eficacia y menor coste [3].

Una de las bacterias más estudiadas por la rama de la biología y la medicina es la *Escherichia coli* (*E. coli*), presente en la microbiota gastrointestinal. Se trata de un microorganismo modelo empleado frecuentemente en estudios experimentales debido a su elevada velocidad de crecimiento y sus pocas necesidades nutricionales. También es utilizada en experimentos de biología y genética molecular, por su facilidad para desplazar el material genético, gracias a la flexibilidad de su genoma [4].

La adición del gen CimA3.7 (originario de *Methanococcus jannaschii*), posibilita que la bacteria *E. coli* sintetice citramalato a partir de piruvato y Acetil-CoA. Para optimizar la producción del citramalato se están desarrollando modelos matemáticos en el campo de la biología de sistemas que representan las reacciones químicas que se observan en del metabolismo de la bacteria. Estos modelos matemáticos tienen como finalidad sustituir los costosos y duraderos experimentos biológicos, utilizados para estudiar los distintos parámetros de operación, por modelos computacionales capaces de proporcionar la información necesaria en cuestión de segundos [1].

1.2 Objetivo

Para poder delegar el trabajo de los métodos experimentales en la ingeniería metabólica, es necesario plasmar el mapa metabólico del microorganismo en un modelo matemático, de manera que este sea capaz de simular su comportamiento a lo largo del tiempo, ante las condiciones establecidas por el usuario.

Los modelos matemáticos existentes acerca del metabolismo de *E. coli* se encuentran en desarrollo. La primera reconstrucción genómica, K-12 MG1655, fue creada en el año 2000, a partir de entonces, el modelo ha sufrido numerosas actualizaciones. Estas redes metabólicas contienen información sobre la esquiotiometría de las reacciones metabólicas, las fórmulas químicas, la carga de los metabolitos y las asociaciones que se llevan a cabo entre los genes, las proteínas y las reacciones [5]. En este trabajo se van a estudiar dos tipos de modelos matemáticos diferentes; un modelo estequiométrico que abarca casi toda la red metabólica de la bacteria, contando con un gran número de metabolitos y reacciones; y un modelo cinético, que contiene una red metabólica mucho más reducida, pero incluye información dinámica sobre las reacciones. La figura 1.1 muestra un esquema de las principales rutas metabólicas de *E. coli*.

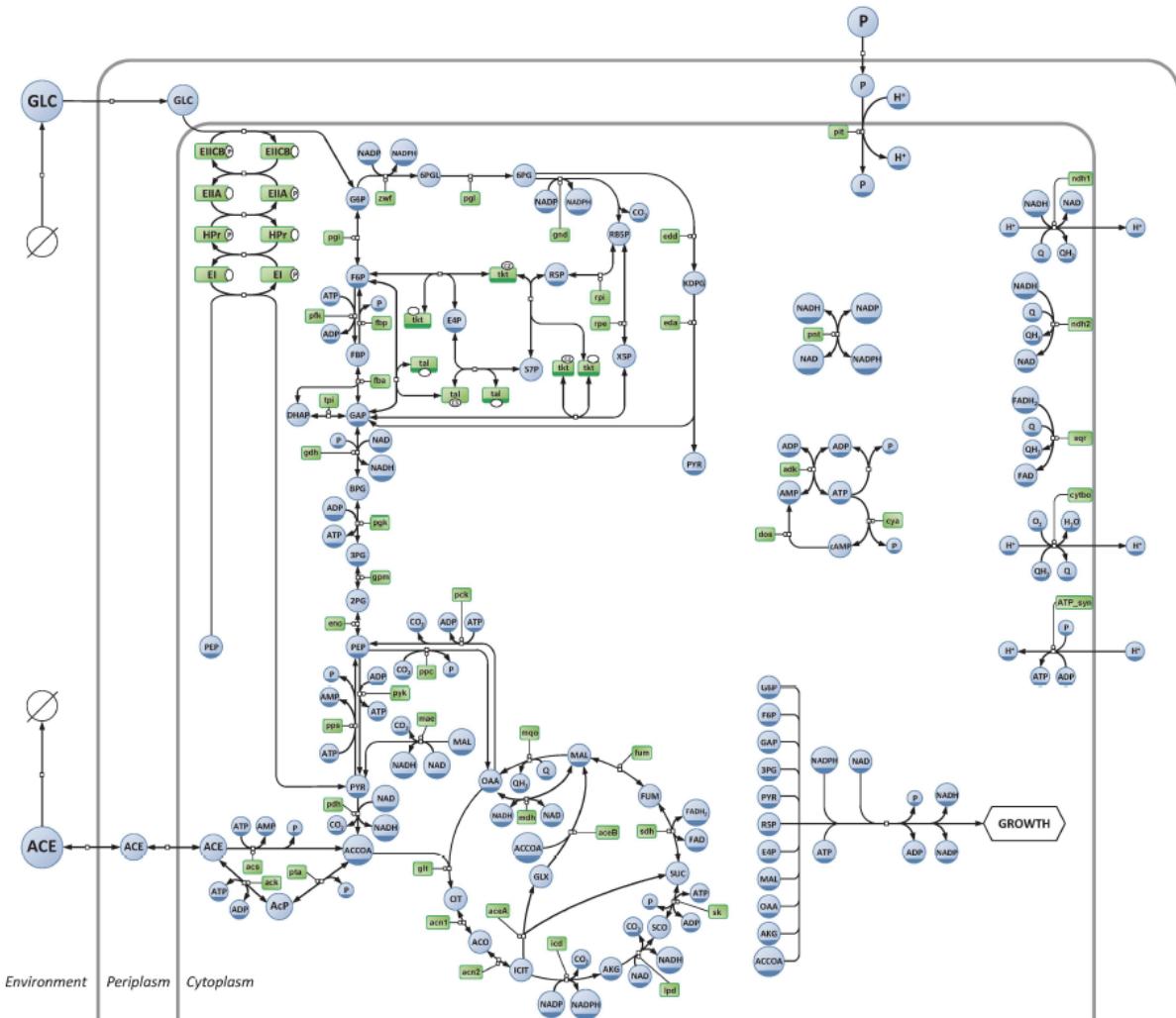


Figura 1.1: **Representación de la red central de carbón y energía de *E. coli*.** Los metabolitos se muestran en azul y las enzimas en verde. [6]

Entre las reacciones del modelo de cinético no se encuentra la síntesis del citramalato, pues la bacteria *E. coli* no sintetiza citramalato de forma natural. Para que produzca citramalato, la bacteria tiene que ser manipulada genéticamente. Por tanto, es necesario añadir la reacción del citramalato al modelo original, reflejando en este modelo rectificado, el comportamiento de la bacteria modificada genéticamente. [7]

El ácido citramalato puede ser obtenido a partir de una reacción metabólica específica, los metabolitos encargados de producir citramalato son el piruvato (PYR) y el Acetil-CoA (ACCOA), que, en presencia de agua (H_2O), dan lugar a hidrógeno (H^+), citramalato y Coenzima A (CoA-SH). La reacción abreviada se muestra en la ecuación (1) y la reacción química en la ecuación (2). [8]



Para optimizar la producción de citramalato se deben considerar varios factores presentes en el desarrollo de la bacteria; siendo necesario encontrar un punto de operación óptimo. Por ejemplo, si se alcanza un flujo de citramalato muy elevado, pero se necesita proporcionar un gran flujo de glucosa, esta situación no será económicamente conveniente. A su vez, a una mayor velocidad de crecimiento, el tamaño de la bacteria aumentará, y se sintetizará una mayor cantidad de los metabolitos de interés por unidad de tiempo, pero esta situación tendría consecuencias negativas en otras variables. La función que relaciona los parámetros a considerar se denomina productividad del citramalato. Cuando la productividad sea máxima, se habrán encontrado los parámetros óptimos de operación que optimizan la producción del citramalato. [1]

La productividad (h^{-1}) en este estudio, tiene en cuenta el crecimiento específico de la bacteria (h^{-1}), la producción de glucosa ($mg\ s^{-1}$) y el flujo de entrada de glucosa al cultivo de bacterias ($mg\ s^{-1}$). El valor proporcionado de crecimiento en la simulación se encuentra en unidades de s^{-1} , pero para facilitar la lectura de los resultados, se transforma en h^{-1} . La ecuación (3) muestra la fórmula que relaciona las variables mencionadas.

$$\text{Productividad (}h^{-1}\text{)} = \text{Crecimiento}(\text{}h^{-1}\text{)} \cdot \frac{\text{Flujo de citramalato (}mg\ s^{-1}\text{)}}{\text{Flujo de glucosa (}mg\ s^{-1}\text{)}} \quad (3)$$

Como en cualquier estudio contemporáneo, los costes y su reducción, deben ser analizados para que el proyecto pueda ser viable en un futuro. En el caso de la ingeniería metabólica, la disminución de costes está basada en la minimización del medio de cultivo, que consiste en minimizar la entrada de nutrientes a la célula sin modificar el valor del resto de variables. A parte de esto, otros puntos a considerar serían el coste de los biorreactores, así como de las materias primas o el mantenimiento de los equipos.

El objetivo del trabajo será combinar la información de dos modelos matemáticos distintos sobre las reacciones del metabolismo de *E. coli*, para calcular las condiciones óptimas de operación que maximizan la productividad del citramalato; considerando la minimización de costes de producción con el fin de poder realizar una aplicación industrial futura.

1.3 Metodología

Los modelos matemáticos existentes de las rutas metabólicas de *E. coli* van a ser analizados y simulados a través de módulos computacionales, para ello se va a utilizar el lenguaje de programación Python en Spyder3, un entorno de desarrollo interactivo. Para maximizar la productividad se utilizan dos modelos metabólicos distintos, un modelo cinético y un modelo estequiométrico, contando cada uno con información complementaria sobre las rutas metabólicas. Combinando ambos modelos se consigue un resultado más completo y preciso sobre las condiciones óptimas de operación. En la figura 1.2 se muestra un esquema global del procedimiento seguido en el proyecto para alcanzar el objetivo definido.

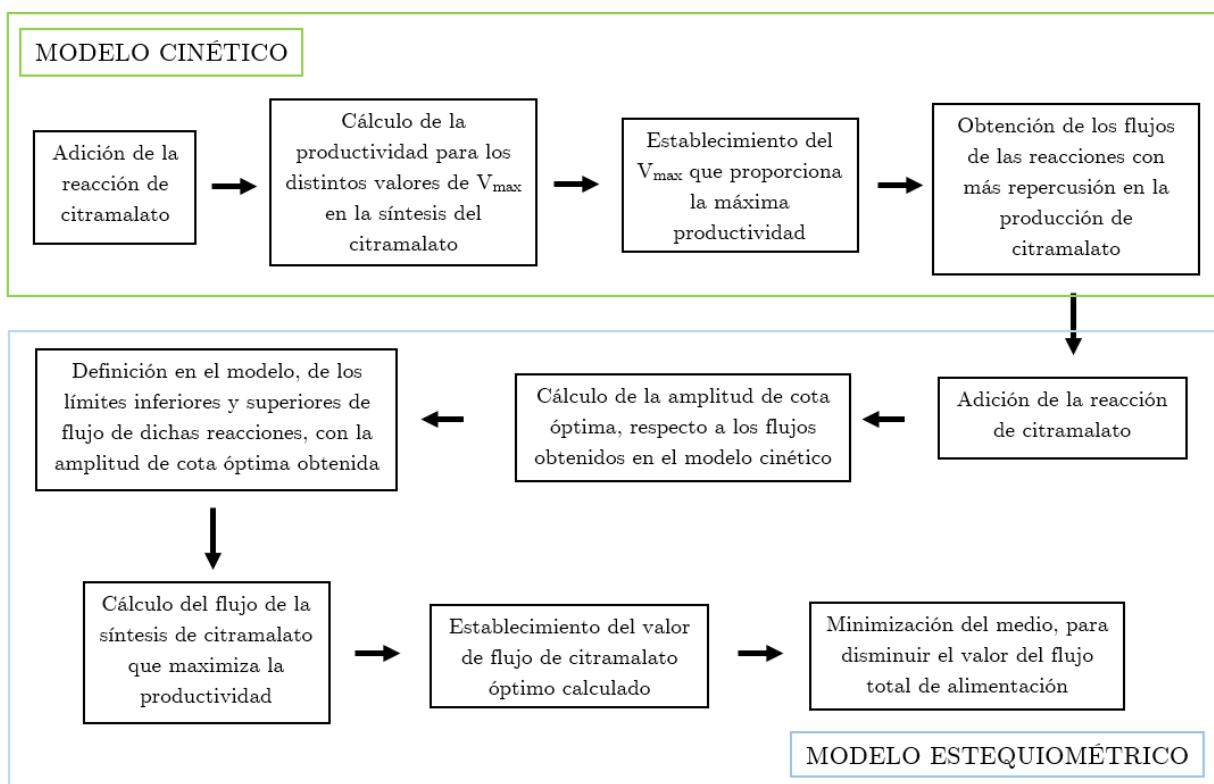


Figura 1.2: *Diagrama de bloques del procedimiento seguido a lo largo del proyecto.*

1.3.1 El estado estacionario

Desde un punto de vista práctico, referido a la producción industrial, interesa producir citramalato de modo continuo; que matemáticamente, equivale al estado estacionario. Este estado se alcanza cuando las concentraciones de los metabolitos y los flujos de las reacciones se mantienen constantes en el tiempo. En el sistema estudiado se traduce como la igualdad entre la cantidad de sustancias que consume

el organismo y la cantidad de productos que salen del sistema, según muestra la ecuación (4).

$$\sum \text{Flujos de entrada} - \sum \text{Flujos de salida} = 0 \quad (4)$$

En el modelo, el estado estacionario está definido de forma que, el mayor ratio de cambio de concentración tiene que ser menor a un umbral previamente definido, para considerar constante el valor de los distintos parámetros. La ecuación (5) muestra la condición establecida para detectar que el modelo ha alcanzado el estado estacionario.

$$\max_{X_i} \left| \frac{d[X_i]}{dt} \right| < 10^{-6} \text{ mM s}^{-1} \quad (5)$$

Siendo $[X_i]$ la concentración de las especies químicas. En este caso, el umbral máximo escogido para alcanzar el estado estacionario ha sido 10^{-6} mM s⁻¹. Cuando se cumple esta condición podemos decir, que la bacteria ha alcanzado el estado estacionario. El tiempo de simulación para el cual se detectaba que los parámetros se mantenían constantes, ha sido 720-000 segundos, equivalente a 200 horas; todas las simulaciones realizadas se han llevado a cabo durante este periodo de tiempo.

1.3.2 Modelo cinético

En 2017, P. Millard [6] desarrolló un modelo cinético de las reacciones metabólicas de la *E. coli*, en el cual la bacteria estaba alimentada por glucosa bajo condiciones aerobias. Es la primera reconstrucción genómica que vincula el metabolismo con el entorno y la proliferación celular a través de los niveles de metabolitos intracelulares. Este modelo contiene tres compartimentos diferentes: extracelular, periplasma y citoplasma. Dentro del modelo hay definidos 62 metabolitos y 68 reacciones, pertenecientes a las principales rutas metabólicas centrales de energía y carbono; sin embargo, estas solo son pequeño porcentaje del total de reacciones que se llevan a cabo en el metabolismo de la bacteria. Esto deriva en una inexactitud a la hora de analizar la producción de los diferentes sustratos, ya que un mismo metabolito puede participar en otras rutas no incluidas en el modelo; esto conlleva a un cierto porcentaje de error en la concentración obtenida.

El primer paso para el análisis del modelo es la adición del metabolito citramalato, así como la definición de su síntesis, para ello, se ha utilizado la librería **libsbml** de Python, capaz de leer, escribir y manipular un modelo almacenado en

un fichero con formato SMBL (Systems Biology Markup Language). Una vez añadido el metabolito, se define la reacción de citramalato. Posteriormente hay que simular el modelo para hallar las condiciones óptimas de operación. La librería `roadrunner`, diseñada para leer, analizar y simular modelos metabólicos; ha sido la utilizada a lo largo de este proyecto para simular el modelo. De aquí se obtiene tanto la velocidad máxima como los flujos de las reacciones más influyentes en la productividad, que maximizan el valor de este parámetro.

1.3.3 Modelo estequiométrico

Para contrastar la pobreza en reacciones del modelo cinético, se analiza un modelo estequiométrico, que cuenta con un total de 1366 metabolitos y 2251 reacciones. [4] El modelo estequiométrico es un modelo matemático que caracteriza los genes y las reacciones de las rutas metabólicas. El modelo carece de información cinética, solo permite establecer un límite superior e inferior para el flujo de la reacción, es decir, definir cotas de flujo de las distintas reacciones. Pudiendo así, establecer unas cotas de flujo, lo más similares posibles a la realidad para poder estudiar la síntesis de los metabolitos en cuestión. [3]

La modificación genómica para incluir la síntesis del citramalato en el metabolismo, se ha realizado con la librería `cobra.py`. El módulo `cobra.py` es un paquete de Python que proporciona una interfaz sencilla para la reconstrucción y el análisis metabólico, basados en los límites de las cotas de flujo [15].

Con este modelo se calculan las cotas de flujo óptimas, respecto a los valores obtenidos en el modelo cinético, y el valor de flujo de citramalato que permite obtener la máxima productividad. Por último, se realiza una minimización del medio de cultivo, para reducir el coste del proceso.

Capítulo 2

Análisis del modelo cinético.

Un modelo cinético contiene información acerca de la velocidad a la que se desarrollan las reacciones químicas definidas en él, en función de las concentraciones de sus metabolitos. El modelo matemático utilizado en este trabajo, “*E.coli_Millard2016.xml*” [6], describe las interacciones entre los metabolitos, así como los valores de los principales parámetros cinéticos: la constante de Michaelis-Menten (K_m) y la velocidad máxima (V_{max}), véase la ecuación (6).

2.1 Adición de la reacción de citramalato.

Para que la bacteria sea capaz de producir citramalato, tiene que ser modificada genéticamente, como se ha explicado en el apartado 1.1. En el modelo cinético, para realizar esta modificación genética, hay que incluir el metabolito citramalato y la reacción que lo sintetiza. Esta adición se lleva a cabo con librería **libSBML**, como se ha explicado en el apartado 1.2.

Para empezar, se crea el metabolito citramalato y posteriormente, se definen los sustratos que intervienen en su síntesis, ya presentes en el modelo; por último se establece la reacción que da lugar al citramalato. Este modelo matemático solo tiene definida la cinética de las reacciones, no tiene información estequiométrica; por tanto, solo hay que fijar los parámetros cinéticos de la reacción: K_m y V_{max} .

La constante K_m tiene un valor de 0,495 mM y V_{max} es inicialmente 4 mM s⁻¹, estos valores han sido obtenidos experimentalmente por investigadores de la Facultad de Ingeniería de la Universidad de Nottingham. La velocidad de la reacción sigue el comportamiento de la ley cinética de Michaelis-Menten, considerando que el piruvato es el reactivo en exceso [7]. La ecuación 4 muestra la ecuación de velocidad para la síntesis del citramalato en *E. coli*.

$$\frac{d[\text{citramalato}]}{dt} = \frac{V_{\max} \cdot [\text{acetyl-CoA}]}{[\text{acetyl-CoA}] + K_m} \quad (6)$$

Dentro del modelo, es necesario definir el valor de los dos parámetros y la ecuación de velocidad que los relaciona. Por último, se escribe sobre el modelo todo lo definido anteriormente, para modificar el modelo ya existente, y que este simule el comportamiento de *E. coli* modificada genéticamente para producir citramalato. Todos los pasos llevados a cabo con libsbml se encuentran en el Anexo A.I y A.II.

2.2 Variación de V_{\max} para maximizar la productividad

El modelo cinético permite calcular la velocidad máxima de la síntesis de citramalato que maximiza su productividad, tras alcanzarse el estado estacionario.

Para conocer el valor de la productividad correspondiente a cada valor de V_{\max} , se realiza una iteración en la que se simula el modelo desde un tiempo 0 hasta 720000 segundos, en la cual se obtienen los valores de la velocidad de crecimiento de la bacteria y el valor del flujo de citramalato; con estos datos, y el flujo de entrada de glucosa, que es constante en todo momento, se calcula la productividad con la ecuación (3). Los resultados obtenidos se guardan en un vector para poder compararlos y realizar una representación gráfica posterior.

Primero, se lleva a cabo una iteración entre unos valores de V_{\max} de 0,1 y 2 mM s⁻¹, donde se calculan sus respectivos valores de productividad. Los resultados obtenidos se presentan en la tabla 2.1.

Tabla 2.1: *Resultados de productividad para cada V_{\max} estudiado.*

V_{\max} (mM s ⁻¹)	Productividad(h ⁻¹)	V_{\max} (mM s ⁻¹)	Productividad(h ⁻¹)
0.1	0.006945	1.1	0.01248
0.2	0.01152	1.2	0.01139
0.3	0.01440	1.3	0.01038
0.4	0.01604	1.4	0.009353
0.5	0.01675	1.5	0.008445
0.6	0.016796	1.6	0.007619
0.7	0.016363	1.7	0.006874
0.8	0.01561	1.8	0.006207
0.9	0.01465	1.9	0.005612
1.0	0.01359	2.0	0.005083

La gráfica 2.1, muestra la representación gráfica de los resultados de la tabla 2.1.

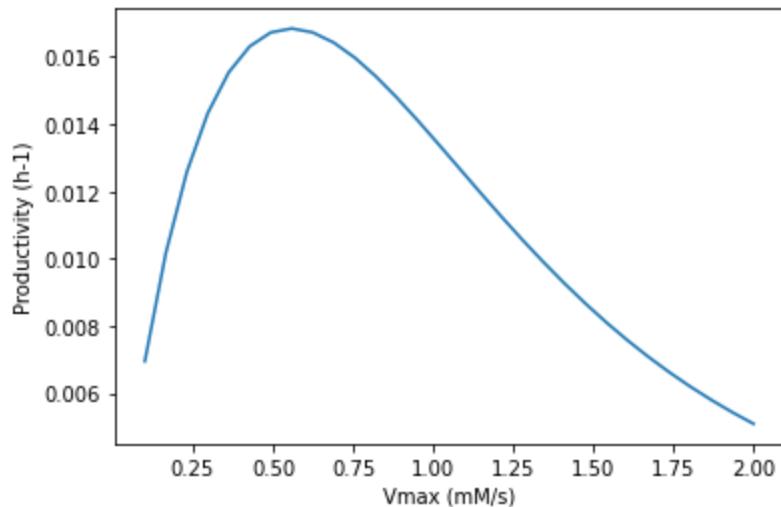


Figura 2.1: **Representación de la productividad frente a V_{max} .** Gráfica obtenida en Spyder3 de la evolución de la productividad con la velocidad máxima de la reacción entre 0.1 y 2 mM s⁻¹.

En esta primera iteración, se observó el máximo de productividad entre los valores 0,5 y 0,6. Para encontrar un valor más preciso de V_{max} se repitió el mismo procedimiento con unos valores entre 0,5 y 0,6. Los valores conseguidos con esta nueva simulación se muestran en la tabla 2.2.

Tabla 2.2: **Resultados de productividad para V_{max} desde 0.5 a 0.6 mM s⁻¹.**

V_{max} (mM s ⁻¹)	Productividad (h^{-1})
0.538	0.01684
0.544	0.01684
0.55	0.01684
0.556	0.01685
0.562	0.01684
0.568	0.01684
0.574	0.01684
0.58	0.01683

El valor de V_{max} , obtenido en la última simulación, que maximiza la productividad es 0.556 mM/s. Por tanto, los parámetros cinéticos óptimos de la síntesis del citramalato ya son conocidos, se establece este nuevo valor como velocidad máxima de la síntesis de citramalato en el modelo para cálculos posteriores.

2.3 Obtención de las cotas de flujo con más influencia en la síntesis del citramalato.

Conociendo los parámetros cinéticos que maximizan la productividad del citramalato, a continuación, se van a obtener el valor de los flujos de las reacciones con más repercusión en la variable a optimizar, con el fin de mejorar la información del modelo estequiométrico.

Considerando su influencia en la productividad del ácido, se han seleccionado un total de 13 reacciones identificadas con las siguientes abreviaciones: XCH_P, PIT, PGI, PFK, FBA, TPI, RPE, MQO, PTA, ACS, ADK, CYA, _ACE_OUT. La relación de estas con la producción de citramalato en la bacteria es explicada a continuación.

El reactivo limitante en la reacción del citramalato es el Acetyl-CoA, los niveles de este metabolito están directamente controlados por las reacciones ACS y PTA [8]. Las reacciones que involucran tanto la producción como el transporte del fósforo tienen un papel crucial en la bacteria, este elemento participa en el metabolismo energético, así como en la formación de macromoléculas clave para la bacteria; las reacciones XCH_P y PIT, regulan la salida y el transporte, respectivamente, de fósforo en la célula [9]. La glucólisis es una de las rutas más importantes y vitales en el microorganismo, ya que proporciona la energía necesaria a la célula, siendo PGI, PFK, TPI y FBA unas de las reacciones principales en este proceso [10]. La reacción ADK es la primera reacción en producir ATP, la molécula que transporta la energía a través de la bacteria [11]. La reacción CYA cataliza la síntesis de cAMP proveniente del ATP. La salida de acetato de la bacteria pertenece a la reacción _ACE_OUT, por otro lado, este metabolito también es uno de los reactivos que da lugar a Acetyl-CoA [6]. La reacción RPE es una de las reacciones previas a la glucolisis, que permiten que se lleve a cabo [12]. Por último, la reacción MQO da lugar a Oxaloacetato, un metabolito precedente al piruvato.

Tras definir las reacciones con más influencia en el proceso, se establece en el modelo el valor óptimo obtenido de velocidad máxima de citramalato (V_{max}), correspondiendo a $0,556 \text{ mM s}^{-1}$; posteriormente se obtienen todos los flujos de las reacciones mencionadas, bajo los parámetros cinéticos definidos. La tabla 2.3, muestra los resultados de la simulación.

Tabla 2.3: *Flujos de las reacciones seleccionadas como más influyentes que proporcionan la máxima productividad.*

Reacción	Flujo (mM s ⁻¹)
XCH_P	0.02847
PIT	0.02847
PGI	0.16798
PFK	0.19018
FBA	0.19006
TPI	0.19006
RPE	0.02263
MQO	2.5722
PTA	0.01117
ACS	0.01117
ADK	0.01911
CYA	0.007930
ACE_OUT	6,345·10 ⁻⁷
CIT	0.09432

El flujo de citramalato calculado es 0,09432 mM s⁻¹. Con estos procedimientos, se han obtenido las condiciones óptimas de operación que maximizan la productividad, teniendo en cuenta la cinética de las reacciones incluidas. Aquí termina el análisis del modelo cinético, del cual se ha extraído el valor de los flujos de las reacciones con más repercusión en la productividad, con el fin de transferir esta información al siguiente modelo a estudiar.

Capítulo 3

Modelo estequiométrico

La multitud de reacciones que se llevan a cabo en el metabolismo de la bacteria se han intentado plasmar en este modelo, a fin de conseguir una simulación análoga a los procesos experimentales. Para mejorar los resultados obtenidos en el capítulo 2, los flujos obtenidos en el análisis anterior van a ser definidos en el modelo estequiométrico, repitiendo la optimización de la producción de citramalato. [4]

3.1. Análisis del balance de flujos.

El cómputo del modelo estequiométrico para la optimización de la productividad de citramalato está basado en análisis del balance de flujo o *Flux Balance Analysis* (FBA). FBA es un método de cálculo del flujo de los metabolitos a través de rutas metabólicas, que permite predecir el ratio de crecimiento de la célula o el flujo de producción de un metabolito en concreto [13].

El principio de este método es la representación matemática de reacciones metabólicas, donde su característica principal es la tabulación en forma de matriz numérica; la matriz contiene los coeficientes estequiométricos de las reacciones del modelo, teniendo en cuenta las restricciones de flujo impuestas. Se establece una función objetivo para determinar la influencia que tienen los flujos de la red en dicha función. En este caso, la función objetivo es la reacción de la biomasa, definida de tal forma que el flujo que la atraviesa, es igual al ritmo de crecimiento exponencial del organismo [13].

Matemáticamente, la función objetivo se usa cuantitativamente para conocer cuando contribuye cada reacción al fenotipo. La representación matemática de las reacciones metabólicas y de la función objetivo define un sistema de ecuaciones lineales que da lugar a un problema de programación lineal. La librería `cobra.py` es

capaz de resolver estos sistemas de ecuaciones, maximizando la función objetivo con los flujos disponibles definidos, mediante las ecuaciones de balance de masa y los límites de las reacciones [12].

Este procedimiento tiene limitaciones, no utiliza parámetros cinéticos por lo que no es capaz de predecir concentraciones, solo permite conocer flujos en estado estacionario.

3.2. Adición de la reacción de citramalato

En primer lugar, se crea el metabolito citramalato (CIT) dentro del compartimento celular, para incluirlo posteriormente en su síntesis, según la reacción mostrada en la ecuación (1). Después, se requiere extraer el citramalato de la bacteria, de lo contrario se acumularía indefinidamente en ella. Dos reacciones de transporte, ambas por difusión, son añadidas al modelo; primero, el citramalato es transportado del interior de la célula al periplasma y posteriormente al exterior; por último, se agrega una reacción de eliminación del citramalato, para que no sea almacenado en el exterior del sistema, sino que se extraiga por completo y el sistema pueda mantener el estado estacionario. La figura 3.1 muestra un esquema del proceso introducido en el modelo.

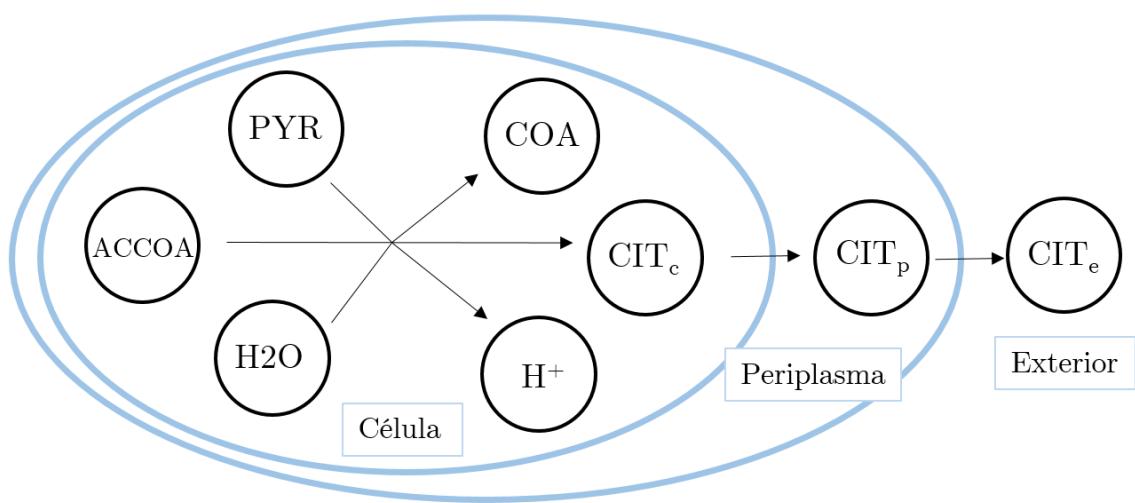


Figura 3.1: *Esquema del proceso de producción de citramalato establecido en el modelo estequiométrico. Siendo los metabolitos presentes piruvato (PYR), Acetil-CoA (ACCOA), agua (H₂O), hidrógeno (H⁺), Coenzima A (COA) y citramalato (CIT_c).*

Se considera que todo el citramalato producido es retirado de la célula, por lo tanto, todas las reacciones de transporte tienen una estequiometría 1:1. El procedimiento utilizado se detalla en el Anexo A.III.

3.3. Establecimiento de las cotas de flujo.

Como se ha explicado en la sección 1.2, el único parámetro variable de las reacciones en el modelo estequiométrico, son las cotas de flujo. Por lo tanto, los valores obtenidos en el apartado 3.1, se van a utilizar para establecer los límites de flujo, intentando que sean los más parecido posible a los obtenido en el anterior análisis para simular las condiciones óptimas obtenidas.

En este modelo, los identificadores de los metabolitos y las reacciones son diferentes a los del modelo estudiado en el Capítulo 2, así como también lo son sus unidades. Las unidades del flujo de las reacciones en el modelo cinético son mM s^{-1} mientras que en el estequiométrico son $\text{mmol g}_{\text{DW}}^{-1}\text{h}^{-1}$. Para llevar a cabo esta conversión de unidades es necesario conocer el volumen de *E. coli*, siendo este $1,77 \text{ L g}_{\text{DW}}^{-1}$ [7]. Con este dato, obtenemos el factor necesario para convertir las unidades del flujo, según el procedimiento desarrollado en la ecuación (7); siendo X un flujo molar cualquiera.

$$(X \text{ mM s}^{-1}) \cdot (1,77 \cdot 10^{-3} \text{ L g}_{\text{DW}}^{-1}) \cdot (3600 \text{ s h}^{-1}) = X \cdot 6,372 \text{ mmol g}_{\text{DW}}^{-1}\text{h}^{-1} \quad (7)$$

Se obtiene un factor de conversión de unidades de 6,372. Como el modelo estequiométrico no tiene información sobre la cinética de las enzimas, los valores asociados al modelo cinético pueden utilizarse en la estructura del modelo estequiométrico sin ningún ajuste en las unidades, solo varían las unidades de las concentraciones de los metabolitos.

Todas las reacciones existentes en el modelo cinético también se encuentran en el modelo estequiométrico, pero poseen una identificación diferente, por tanto, se requiere hallar el identificador equivalente en este segundo modelo. La tabla 3.1 muestra las equivalencias de los nombres de las reacciones de los dos modelos.

Tabla 3.1: *Nomenclatura de las reacciones equivalentes en los dos modelos.*

Nomenclatura del modelo cinético	Nomenclatura del modelo estequiométrico
XCH_P	PItex
PIT	PIt2rpp
PGI	PGI
PFK	PFK
FBA	FBA
TPI	TPI
RPE	RPE
MQO	MDH2
PTA	PTAr
ACS	ACS
ADK	ADK1
CYA	ADNCYC
ACE_OUT	ACtex

Tras localizar todas las reacciones en el modelo, se procede a obtener el valor de las cotas de flujo que maximiza la productividad. El valor de esta cota hace referencia al rango que se establece, dentro del cual se deben encontrar los flujos de las reacciones. Para optimizar este valor, se realiza una iteración en la que se establece un límite inferior y un límite superior de cota en todas las reacciones, con un porcentaje de amplitud menor y mayor respectivamente, al flujo hallado en modelo cinético. Hay dos excepciones: la alimentación de glucosa que es constante y posee el mismo valor que en el modelo cinético, $0,23 \text{ mM s}^{-1}$ o $1,456 \text{ mmol/gDW}^{-1} \text{ h}^{-1}$; y la reacción de salida del acetato, que al obtener un valor muy próximo a 0, es necesario proporcionar un poco más de amplitud al flujo para que el modelo pueda ser optimizado, en este caso la cota inferior establecida es $-0,1 \text{ mmol/gDW}^{-1} \text{ h}^{-1}$ y la superior $0,1 \text{ mmol/gDW}^{-1} \text{ h}^{-1}$. Las ecuaciones (8) y (9) muestran las ecuaciones determinadas en el programa para definir dichos límites.

$$\text{Límite inferior} = \text{flujo} \cdot (1 - \text{amplitud de cota}) \quad (8)$$

$$\text{Límite superior} = \text{flujo} \cdot (1 + \text{amplitud de cota}) \quad (9)$$

Siendo flujo ($\text{mmol/gDW}^{-1} \text{ h}^{-1}$) el flujo de las distintas reacciones estudiadas.

Se ha estudiado como varía la productividad del citramalato con los límites de flujo. Los valores de amplitud estudiados van desde el 0% al 40 %. La gráfica 5 muestra cómo evoluciona la productividad conforme varía la amplitud de cota.

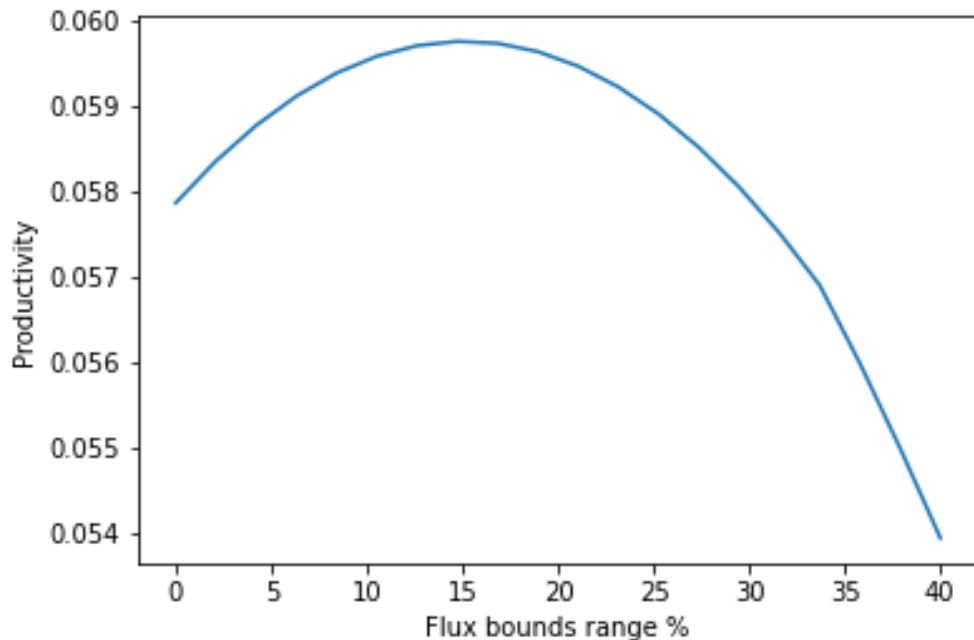


Figura 3.2: *Evolución de la productividad con el aumento de la amplitud de la holgura de cota.*

Cuando la amplitud es 0, las condiciones definidas son las mismas que las proporcionadas por el modelo cinético en condiciones de máxima productividad; no obstante, este modelo cuenta con un mayor número de reacciones; por consiguiente, las condiciones para obtener la máxima productividad pueden variar. Como muestra la gráfica 3.2, la productividad aumenta con la amplitud de flujo hasta que se alcanza el 14,737% que empieza a disminuir, siendo este valor, la amplitud de cota que facilita el máximo valor de productividad.

Posteriormente se establece este porcentaje de amplitud de cota en todas las reacciones mencionadas, para cálculos posteriores.

3.3 Análisis de la influencia del flujo de citramalato en la productividad.

Una vez están definidos los límites de cota de las reacciones mencionadas, se va a proceder a estudiar la influencia del flujo de citramalato en su productividad, con

el fin de calcular el valor del flujo del ácido correspondiente al punto de mayor productividad.

El análisis de la variación del flujo de citramalato se estudia estableciendo unos límites superior e inferior de flujo que lo obliguen a converger a un mismo valor, es decir, que ambos límites sean iguales. Por tanto, se modifican estos valores, desde 0 hasta 12 $\text{mmol gDW}^{-1} \text{ h}^{-1}$, realizando una iteración en la que, se establezca un flujo de citramalato, se obtenga el valor de la velocidad de crecimiento de la bacteria y posteriormente el valor de la productividad. Tanto los valores de flujo utilizados como los resultados de crecimiento y productividad han sido almacenados en vectores. Las gráficas 3.3 y 3.4 muestran la variación del crecimiento y la productividad, respectivamente, con el flujo de citramalato.

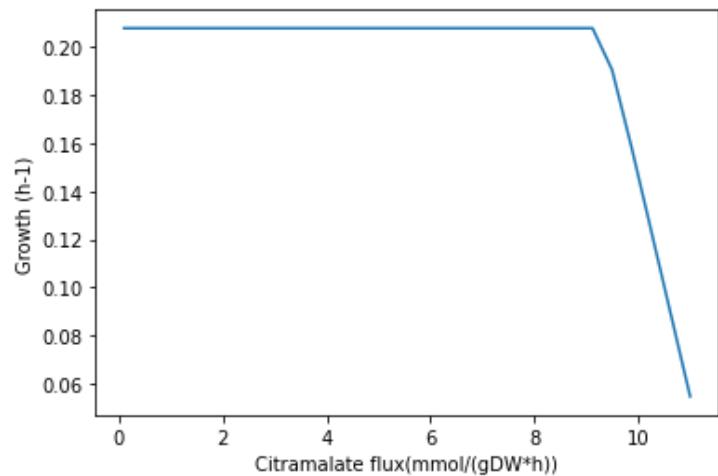


Figura 3.3: *Evolución de la velocidad de crecimiento con la variación de flujo de citramalato.*

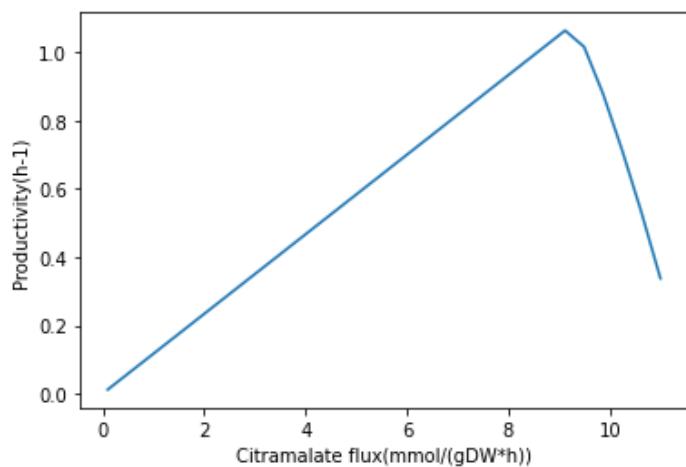


Figura 3.4: *Evolución de la productividad con la variación de flujo de citramalato.*

El valor de flujo de citramalato correspondiente al punto de mayor productividad es $9,1207 \text{ mmol gdw}^{-1}\text{h}^{-1}$. Como se puede observar en la gráfica 3.3, el crecimiento de la bacteria tiene un ritmo constante, a pesar del aumento de la síntesis de citramalato, hasta que se alcanza a un valor de flujo del ácido de $9,1207 \text{ gdw}^{-1}\text{h}^{-1}$ en el que la velocidad crecimiento cae bruscamente. Esto se debe a que la síntesis de citramalato hace uso cada vez, de más cantidad de reactivos siendo estos eliminados de otras reacciones; por tanto, llega un punto en el que otras rutas fundamentales para el crecimiento de la célula, no cuentan con todos los sustratos necesarios, y la célula no es capaz de llegar al tamaño alcanzado inicialmente.

Comparando este resultado con el calculado en el modelo cinético podemos observar un notable aumento en la producción de citramalato. Convirtiendo las unidades del modelo estequiométrico al modelo cinético con el factor explicado en la ecuación (6), conseguimos un flujo de citramalato de $1,4314 \text{ mM s}^{-1}$, que comparado con el hallado en el capítulo 2, siendo este $0,09432 \text{ mM s}^{-1}$, se ha conseguido una notable mejora en su producción. El flujo actual es aproximadamente quince veces mayor que el obtenido en el anterior modelo.

3.4. Minimización del medio de cultivo

En este trabajo, se define el medio mínimo de cultivo como el ambiente de desarrollo de un organismo con las mínimas cantidades posibles de nutrientes que posibilita el crecimiento de la bacteria y la síntesis de citramalato a máxima productividad. Esto se traduce como la minimización de la cantidad de nutrientes que entran a la célula, sin modificar su ritmo de crecimiento inicial, ni tampoco en este caso, el parámetro a optimizar: la productividad. El principal interés de este método es la reducción de los costes de operación [14].

La librería ya mencionada **cobra**, cuenta con la función **minimal_media**, que por defecto obtiene el medio con el aporte de flujo total mínimo a través del análisis de balance de flujo. Para utilizar esta función hay que proporcionar al programa el modelo a optimizar y la función objetivo. La función objetivo es la misma que en el apartado 3.3., la producción de biomasa.

El aporte de flujo está modelado por las reacciones de intercambio del modelo, los metabolitos entran y salen de la bacteria a través de transporte por difusión. Se va a optimizar el flujo de todas las reacciones de intercambio presentes. La tabla 3.2 muestra los flujos de los elementos que entran y salen de la célula, antes y después de computar la función que minimiza el medio.

Tabla 3.2: *Flujo de las reacciones de intercambio inicialmente y después de minimizar el medio de crecimiento de la bacteria.*

Elemento	Flujo inicial (mmol gDW ⁻¹ h ⁻¹)	Flujo optimizado (mmol gDW ⁻¹ h ⁻¹)
Calcio	-0,0011	-0,0011
Cianocobalamina	0	0
Dióxido de carbono	5,0258	3,2916
Cloro	-0,0011	-0,0011
Cobalto	0	0
Cobre	-0,0001	-0,0001
Hierro (+2)	-0,0212	-0,0033
Hierro (+3)	0	0
Glucosa	-10	-10
Hidrógeno (H+)	20,3942	22,6225
Agua	20,0470	17,9030
Potasio	-0,0406	-0,0406
Magnesio	-0,0018	-0,0018
Manganoso	-0,0001	-0,0001
Molibdato	0	0
Sodio	0	0
Amonio	-2,3168	-2,2455
Níquel	-0,0001	-0,0001
Oxígeno	-9,0750	-8,5938
Fósforo	-0,2108	-0,2005
Selenato	0	0
Selenita	0	0
Sulfato	-0,0524	-0,0524
Tungstato	0	0
Cinc	-0,0001	-0,0001

Los elementos que tienen flujos negativos entran a la célula, pertenecen al grupo de los nutrientes; mientras que los flujos positivos son los productos que son extraídos del organismo. Con esta optimización se ha conseguido disminuir la entrada del ion hierro (+2), así como la del ion amonio y la del oxígeno, a la vez que se ha reducido la producción de CO₂, un punto a considerar en cuanto a la contaminación del proceso.

Todos los elementos mostrados en la tabla 3.1 están presentes en el metabolismo de la bacteria, pero algunos de ellos tienen flujo nulo, por lo tanto, podrían ser prescindibles para el sistema. Este es el siguiente punto a tratar, se va a analizar si la eliminación de los distintos compuestos afecta a la productividad del citramalato.

La librería **cobra.py** dispone de una función capaz de minimizar el número de componentes presentes, sin embargo, el *solver* utilizado no ha logrado encontrar la solución tras dos horas de cálculo. Existe una alternativa subóptima para minimizar el, que consiste en escribir manualmente los comandos que lleva incorporados la función. La eliminación se va a llevar a cabo forzando a cero cada uno de los flujos de transporte de los metabolitos. Para ello se ha realizado una iteración en la que se establece el flujo de una reacción a cero, se optimiza el modelo y se calcula la productividad. Los elementos que no han tenido ninguna influencia en la productividad al ser eliminados han sido la cianocobalamina, el hierro (+3), el sodio, el selenato, el selenito y tungstato. Tanto el molibdato como el cobalto, tienen efectos sobre la productividad, independientemente de que su flujo sea nulo. La tabla 3.3, muestra los elementos necesarios en la alimentación y sus respectivos flujos para obtener la máxima productividad.

Tabla 3.3: *Flujos optimizados de entrada para maximizar la productividad de citramalato.*

Elemento	Flujo optimizado
Calcio	-0,0011
Cloro	-0,0011
Cobalto	0
Cobre	-0,0001
Hierro (+2)	-0,0033
Glucosa	-10
Potasio	-0,0406
Magnesio	-0,0018
Manganoso	-0,0001
Molibdato	0
Amonio	-2,2455
Níquel	-0,0001
Oxígeno	-8,5938
Fósforo	-0,2005
Sulfato	-0,0524
Cinc	-0,0001

La alimentación que maximiza la productividad de citramalato, tras haber realizado una minimización del medio cuenta con 16 elementos, los cuales suman un flujo total de $21,1405 \text{ mmol GDW}^{-1}\text{h}^{-1}$, mientras que inicialmente se tenían 22 metabolitos y un flujo de $21,7212 \text{ mmol GDW}^{-1}\text{h}^{-1}$.

Por último, se ha realizado un estudio de la importancia del resto de nutrientes en la productividad del ácido. Se ha repetido el mismo proceso y se han forzado a cero, cada uno de los flujos de todos los metabolitos presentes en el flujo de alimentación. Cabe la posibilidad, de que la función de alguno de los nutrientes sea reemplazable por otro metabolito, y sea factible eliminar este compuesto de la alimentación, aumentando el flujo de otro. Sin embargo, los resultados obtenidos muestran que ninguno de los metabolitos presentes en la alimentación pueden ser eliminados del modelo, todos son necesarios para que el valor obtenido de la productividad no disminuya.

3.5 Condiciones anaerobias.

Los organismos pueden trabajar en condiciones aerobias, en presencia de oxígeno, o en condiciones anaerobias, sin oxígeno. Como podemos deducir de los flujos de entrada, el modelo de *E. coli* se encuentra en condiciones aerobias, pero se puede estudiar el comportamiento de la bacteria en un estado anaerobio, para ello el flujo de oxígeno tendría que ser forzado a cero. Se ha realizado esta verificación, no obstante, el modelo no se ha podido ser optimizado; por tanto, esta reconstrucción genómica no está diseñada para trabajar en un entorno sin oxígeno [17].

Capítulo 4

Resultados

En esta sección se van a resumir y desarrollar los resultados obtenidos a lo largo del proyecto. Se ha combinado la información cinética y estequiométrica presente en los dos modelos analizados, para encontrar las condiciones que maximizan la producción de citramalato mediante la bacteria *E. coli*.

4.1. Parámetros cinéticos

El modelo cinético aporta información sobre la velocidad de las distintas rutas metabólicas, así como de las constantes cinéticas de sus reacciones que permiten conocer las concentraciones de cada metabolito en un instante concreto. En el análisis de este modelo, se han optimizado los parámetros para la producción de citramalato.

La velocidad máxima de la síntesis de citramalato ha sido optimizada considerando la productividad del ácido buscado. Como se ha mencionado en el apartado 2.2, el valor más adecuado es $0,556 \text{ mM s}^{-1}$, con esta velocidad máxima se alcanza el máximo de productividad, obteniendo un flujo de citramalato de $0,09432 \text{ mM s}^{-1}$. Este factor depende proporcionalmente tanto del flujo de citramalato como del crecimiento de la bacteria. La alimentación de glucosa es constante, no se modifica ni varía a lo largo del proceso, por lo que no modifica el valor de la productividad.

Como es lógico, al aumentar la velocidad máxima de la síntesis del citramalato, aumenta el flujo de producción de este metabolito; sin embargo, la velocidad de crecimiento del microorganismo disminuye. Esta disminución se puede justificar por la falta de metabolitos para el desarrollo de la bacteria, qué, conforme aumenta la velocidad máxima de la reacción del ácido, son destinados a su síntesis, y ya no pueden ser empleados en otras reacciones. El piruvato y la Coenzima A, los dos

reactivos principales para la obtención de citramalato; sin embargo, también tienen un papel esencial en el crecimiento de la bacteria, lo que provoca que haya que dividir los metabolitos entre ambos procesos. [18]. Este es uno de los principales motivos para la creación del factor de productividad, ante posibles cambios en la velocidad máxima de la síntesis de citramalato, se obtienen respuestas opuestas en estos dos parámetros. Este es el motivo por el que la gráfica 3.2, que representa la evolución de la productividad con la velocidad máxima de la reacción, presenta una forma cóncava, los dos parámetros se contrarrestan.

4.2. Flujos óptimos para el sistema

Una vez que se conocen los parámetros cinéticos que maximizan la productividad, se han obtenido los flujos de las 13 reacciones seleccionadas con el fin de trasladar esta información al modelo estequiométrico. Se han trasladado los flujos obtenidos en el modelo cinético al modelo estequiométrico, con el fin de establecer un sistema de desarrollo de la bacteria, lo más parecido posible al que proporciona las condiciones óptimas en el anterior modelo. Pero para que esta última reconstrucción genómica pueda mejorar los resultados obtenidos, se ha proporcionado una cierta amplitud de cota, a fin de poder maximizar la productividad de citramalato bajo los parámetros de este nuevo modelo.

Los resultados obtenidos, indican que, al variar los límites de flujo, la productividad es máxima cuando se proporciona una amplitud de cota del 14,373%, respecto a los flujos obtenidos en el modelo cinético. Esto quiere decir que los flujos de las reacciones para este segundo modelo, son muy parecidos a los del modelo anterior, pero debido a la existencia de un mayor número de reacciones o a otros factores, difieren ligeramente y es necesario facilitar una cierta flexibilidad al sistema. La gráfica 6 tienen forma convexa, alcanzando su máximo al 14,373%. Esta situación es análoga a la del apartado 4.1, el crecimiento de la bacteria y el flujo de citramalato tienen respuestas contrarias, se requiere encontrar el punto en el que la multiplicación de ambos factores sea máxima.

En estos últimos cálculos se excluyen la glucosa y el acetato, que son el nutriente y el producto principal respectivamente. Para la alimentación de glucosa, se ha fijado un valor constante para todo el estudio; mientras que, para el acetato, en el modelo cinético se ha obtenido un flujo de este metabolito muy cercano a 0, por lo que se ha dado una amplitud mayor a sus cotas de flujo.

Para la optimización del flujo de citramalato se ha empleado el mismo procedimiento, obteniendo un flujo máximo de $1,4314 \text{ mM s}^{-1}$, equivalente a 1164,59

g L⁻¹h⁻¹ del ácido. Para las cotas de flujo establecidas, se ha ido aumentando la producción de citramalato, al principio la velocidad de crecimiento se mantiene constante hasta que se alcanza un punto en el que el valor de crecimiento cae bruscamente, y por consecuente también lo hace la productividad, este punto es que proporciona el valor máximo de productividad. Al contrario que en el modelo cinético, el crecimiento en esta reconstrucción es capaz de mantenerse constante a pesar del aumento de la velocidad de síntesis de citramalato, esto se puede explicar por la existencia de un mayor número de rutas metabólicas capaces de proporcionar nutrientes suficientes para el desarrollo de la bacteria, a pesar de arrebatar a este proceso, reactivos necesarios para la síntesis del citramalato. En este modelo la bacteria es capaz de crecer utilizando otros metabolitos. Contrariamente, en el modelo cinético, el piruvato y la Coenzima A, son dos metabolitos imprescindibles para su desarrollo, como podemos observar en la evolución de ambas variables.

4.3. Minimización del medio.

El último paso realizado ha sido la reducción del flujo total de alimentación, así como del número total de nutrientes. Tras este cálculo, se ha conseguido reducir el valor total de flujo un 0,5807 mmol gdw⁻¹h⁻¹, lo que equivale a 0,09113 mM s⁻¹. Parece una cifra insignificante, pero en cantidades industriales puede ser un gran avance, ya que si convertimos este último valor a horas obtenemos un ahorro de 328,0791 mmol L⁻¹h⁻¹. Por ejemplo, si se está utilizando un biorreactor de 5L, se habría conseguido una disminución de 1,6405 mol h⁻¹ de alimentación al reactor.

La reducción del flujo total de entrada, se debe la disminución de flujo de tres metabolitos: el oxígeno, el amonio y el fósforo, el valor del resto no se ha sufrido ninguna alteración. Tras la minimización del flujo total, se han comprobado que elementos pueden ser eliminados del grupo de nutrientes sin que se produzca ningún cambio en el valor de la productividad. Se han conseguido eliminar un total de 6 elementos presentes inicialmente en el sistema, innecesarios para la producción de ácido citramalato. La gráfica 4.1 muestra la composición de la alimentación óptima de la bacteria para la producción de citramalato.

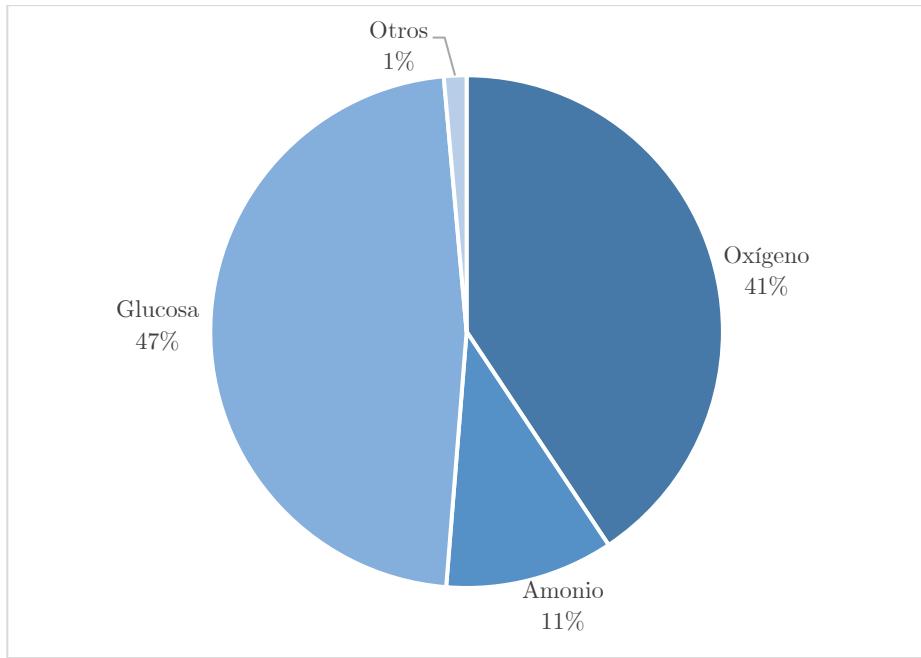


Figura 4.1: *Composición del flujo de alimentación a la bacteria E. coli. Después de llevar a cabo la minimización del medio.*

Como se observa en la gráfica, los dos nutrientes principales, son la glucosa y el oxígeno, seguidos del amonio. El 1% restante está formado por los otros 13 metabolitos que entran a la bacteria: calcio, cloro, cobalto, cobre, hierro (+2), potasio, magnesio, manganeso, molibdato, níquel, fósforo, sulfato y cinc.

Se ha realizado también un estudio de las consecuencias de eliminar cualquiera de estos 16 elementos del flujo de alimentación en la productividad, pero todos son indispensables para mantener el valor de productividad de citramalato, la eliminación de cualquiera de ellos hace que el flujo del ácido sea nulo.

Capítulo 5

Planificación temporal y presupuesto

5.1 Planificación temporal del proyecto

El proyecto ha ocupado un total de 9 meses de trabajo, desde septiembre de 2020 hasta junio de 2021. Cabe destacar que el número de horas dedicadas al proyecto es notablemente más elevado en los últimos 3 meses, abarcando aproximadamente el 60% del tiempo dedicado. Al comienzo de trabajo se realizó un periodo de familiarización con las herramientas a utilizar, en este caso Python y posteriormente las librerías `roadrunner`, `libsbml` y `cobra.py`. Tanto el lenguaje de programación como los instrumentos nombrados eran totalmente desconocidos. Dentro del tiempo dedicado a la simulación del modelo cinético se incluye el periodo de familiarización con las distintas librerías. La tabla 5.1 presenta los 5 bloques principales en los que se ha dividido el proyecto, así como su distribución temporal. Y en la imagen 5.1 se muestra el diagrama de Gantt del trabajo completo.

Tabla 5.1: *Distribución temporal del proyecto*

TAREAS	INICIO	FINALIZACIÓN	DURACIÓN
Simulación del modelo cinético	1-Oct	1-Dec	60 h
Ánálisis de los parámetros cinéticos	1-Dec	1-Feb	30 h
Simulación del modelo estequiométrico	1-Feb	1-Mar	30 h
Ánálisis de los flujos y del medio	1-Mar	15-Apr	50 h
Memoria	1-May	20-Jun	150 h

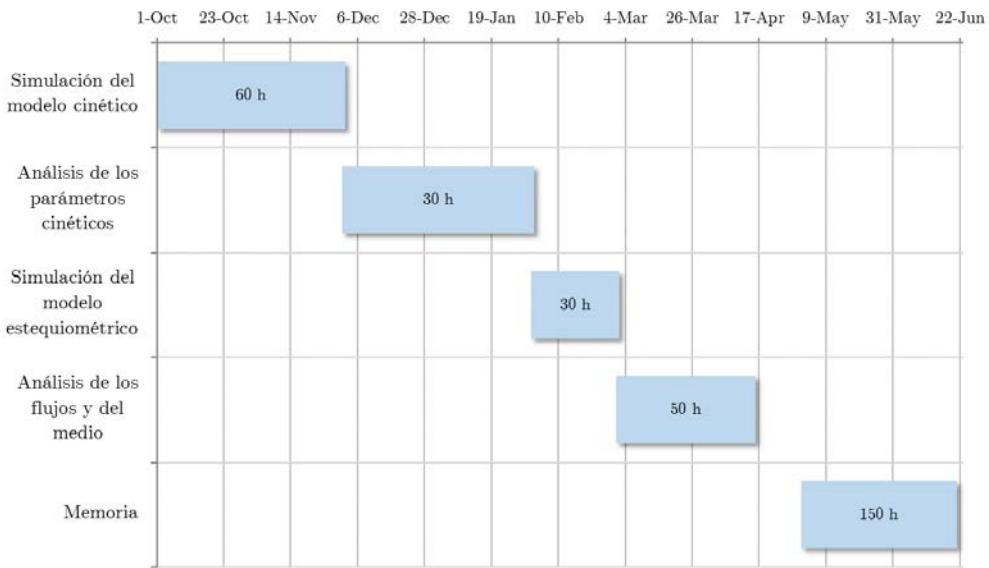


Figura 5.1: **Diagrama de Gantt del proyecto.** Realizado sobre una plantilla de Smarsheet [19]

En total, se han empleado 310 horas para la realización del Trabajo de Fin de Grado, de las cuales la mitad han sido destinadas al cálculo y obtención de los resultados mostrados y la otra mitad a la elaboración de la memoria.

5.2 Estimación de costes

Para estimar el precio total del proyecto se han tenido en cuenta todos los medios utilizados. El salario medio se ha estimado a 10 €/h, teniendo en cuenta que se trata de un estudiante de ingeniería de último curso. El salario del profesor ha considerado 30 €/h. Después a la única herramienta utilizada para el desarrollo ha sido un ordenador portátil, comprado en 2017 por 990€; se ha considerado una amortización lineal y una vida útil de 7 años, de aquí deducimos un coste del proyecto de 141 euros. El portátil, se ha estimado que consume 0,15 kW/h y el precio medio de la electricidad se ha aproximado a 0,200 €/kWh [20]. €

Tabla 5.2: **Estimación de costes del proyecto**

Concepto	Precio (€/h)	Horas (h)	Coste (€)
Estudiante	10	310	3100
Profesor	30	50	1500
Electricidad	0,03	310	9,3
Ordenador	-	-	141
TOTAL :			4750,3 €

De la tabla 5.2, se obtiene un presupuesto total del trabajo de 4750,3 €.

Capítulo 6

Conclusiones

6.1 Conclusiones generales

A lo largo del trabajo se han analizado dos modelos matemáticos sobre las reacciones químicas de las rutas metabólicas de la bacteria Escherichia Coli. Se han desarrollado distintos métodos computacionales para la obtención de los parámetros óptimos de operación, que proporcionan un entorno adecuado en el organismo para la producción de citramalato, tras ser modificado genéticamente para este fin.

En el desarrollo del proyecto se han utilizado distintas librerías pertenecientes al lenguaje de programación Python, utilizado dentro del sistema de operación Linux (distribución Ubuntu), capaces de leer y simular modelos biológicos. Se han desarrollado numerosas iteraciones para hallar el valor máximo de la productividad ante la variación de las distintas variables.

Primero, se han analizado los parámetros cinéticos, obteniendo la velocidad máxima de la reacción de citramalato, que proporciona la máxima productividad. El inconveniente de este resultado ha sido el bajo flujo de producción de citramalato conseguido. Después, en el modelo estequiométrico se ha analizado la influencia de las distintas cotas de flujo, partiendo de un flujo de referencia del modelo cinético que maximizaba la productividad. También se ha estudiado la repercusión del flujo total de alimentación, así como de los distintos nutrientes presentes.

Durante el proyecto se ha demostrado que es posible analizar las condiciones óptimas de operación, en microorganismos biológicos modificados genéticamente, a través de métodos matemáticos mediante simulaciones computacionales, logrando disminuir notablemente el tiempo dedicado a estos estudios. Los resultados obtenidos exponen la viabilidad teórica del objetivo presentado. Se ha calculado un flujo de citramalato suficiente para realizar posibles aplicaciones industriales futuras, en procesos bioquímicos de la obtención de ácido metacrílico.

6.2 Trabajos futuros

Los cálculos realizados en este trabajo se han desarrollado sobre dos reconstrucciones genómicas ya existentes. Los parámetros estudiados son tan solo una pequeña parte de todos los factores que pueden influir en el desarrollo de la bacteria, así como en la síntesis del citramalato. Las posibles líneas de mejoras futuras que se podrían realizar sobre el proyecto se enumeran a continuación:

- Realizar una reconstrucción genómica que combine la información de ambos modelos, es decir, que contenga tanto los datos cinéticos como estequiométricos de las reacciones para poder realizar estudios más precisos.
- Hacer uso de otro tipo de librerías para simular estos modelos con una mayor capacidad, aptas para obtener la solución de los cálculos que no han sido posibles durante este trabajo.
- Estudiar las condiciones óptimas para la producción de citramalato considerando un entorno anaerobio.
- Analizar la influencia de otros factores, como la temperatura o la acidez del medio, en la productividad del citramalato.
- Aplicar la metodología desarrollada a la producción de otros metabolitos.

Capítulo 7

Bibliografía

- [1] Xianghao W. & Mark A, Eiteman (2016). Production of Citramalate by Metabolically Engineered *Escherichia coli*. *Biotechnology and Bioengineering*, Vol. 113, No. 12.
- [2] Nagai K. (2001). New developments in the production of methyl methacrylate. *Appl Catal A-Gen* 221:367–377.
- [3] Wu X., Eiteman MA. (2016). Production of citramalate by metabolically engineered *Escherichia coli*. *Biotechnology and Bioengineering*, Vol. 113, No. 12: 2670-267
- [4] Orth J. D., Conrad T. M., Na J., Lerman J. A., Nam H., Adam M Feist A. M. and Palsson, B. (2011) A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism. *Molecular Systems Biology* 7: 535
- [5] Desmarchelier P. and Fegan N. (2002). *ESCHERICHIA COLI*. *Encyclopedia of Dairy Sciences*: 948-954.
- [6] Millard P., Smallbone K., and Mendes P. (2017). Metabolic regulation is sufficient for global and robust coordination of glucose uptake, catabolism, energy production and growth in *Escherichia coli*. *PLoS Comput Biol* 13(2).
- [7] Wongprommoon A (2018). Optimising production of citramalate based on the *E. coli* kinetic and stoichiometric models. *University of Cambridge*.
- [8] Kadir TA, Mannan AA, Kierzek AM, McFadden J, Shimizu K (2010). Modeling and simulation of the main metabolism in *Escherichia coli* and its several single-gene knockout mutants with experimental verification. *Microb Cell Fact*. 2010; 9:88
- [9] Stigter KA, and Plaxton WC (2015). Molecular Mechanisms of Phosphorus Metabolism and Transport during Leaf Senescence. *Plants (Basel)*. Dec; 4(4): 773–798.

- [10] Ranjbar S., Shahmansouria M., Attribc P., Bogaertsb A. (2020) Effect of plasma-induced oxidative stress on the glycolysis pathway of *Escherichia coli*. *Computers in Biology and Medicine*: Volume 127.
- [11] Satoh T., Kato J., Takiguchi N., Ohtake H., Kuroda A. (2004). ATP Amplification for Ultrasensitive Bioluminescence Assay: Detection of a Single Bacterial Cell. *Biosci. Biotechnol. Biochem.*, 68 (6): 1216–1220.
- [12] Peng L., MJ. Arauzo-Bravo M.J, Shimizu K. (2004). Metabolic flux analysis for a ppc mutant *Escherichia coli* based on ¹³C-labelling experiments together with enzyme activity assays and intracellular metabolite measurements. *FEMS Microbiology Letters* 235: 17–23
- [13] Orth JD, Thiele I, and Palsson BØ (2010). What is flux balance analysis? *Nat Biotechnol*; 28(3): 245–248.
- [14] Soares NC., Spat P., Krug K., Macek B. (2013). Global Dynamics of the *Escherichia coli* Proteome and Phosphoproteome During Growth in Minimal Medium. *Journal of Proteome Research* 12 (6), 2611-2621
- [15] The cobrapy core team (2019). Cobra Documentation. *Cobra*. Release 0.15.4.
- [16] Ebrahim A., Lerman JA, Palsson B.O., Hyduke D.R. (2013) COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Systems Biology* Volume 7: 74.
- [17] Sanchez AM, Bennett GN, San KY. (2005) Novel pathway engineering design of the anaerobic central metabolic pathway in *Escherichia coli* to increase succinate yield and productivity. *Metab Eng* 7:229–239.
- [18] Abdel-Hamid AM., Attwood MM., Guest JR. (2001). Pyruvate oxidase contributes to the aerobic growth efficiency of *Escherichia coli*. *Microbiology*. Volume 147: Issue 6
- [19] Smartsheet (2021). *PLAN DEL PROYECTO Y PLANTILLA DEL GRÁFICO DE GANTT: IC-FL-Gantt-Chart-Excel-ES-27035*.
- [20] Selectra. (2021). Tarifas gas y luz: Consulta el precio diario de la luz (€/kWh): tarifas y comparativa.

Lista de Figuras

1.1	<i>Representación de la red central de carbón y energía de E. coli. Los metabolitos se muestran en azul y las enzimas en verde.</i>	3
1.2	<i>Diagrama de bloques del procedimiento seguido a lo largo del proyecto.</i>	5
2.1	<i>Representación de la productividad frente a V_{max}. Gráfica obtenida en Spyder3 de la evolución de la productividad con la velocidad máxima de la reacción entre 0.1 y 2 $mM s^{-1}$.</i>	10
3.1	<i>Esquema del proceso de producción de citramalato establecido en el modelo estequiométrico. Siendo los metabolitos presentes piruvato (PYR), Acetil-CoA (ACCOA), agua (H₂O), hidrógeno (H⁺), Coenzima A (COA) y citramalato (CITc).</i>	14
3.2	<i>Evolución de la productividad con el aumento de la amplitud de la holgura de cota.</i>	17
3.3	<i>Evolución de la velocidad de crecimiento con la variación de flujo de citramalato.</i>	18
3.4	<i>Composición del flujo de alimentación a la bacteria E. coli. Después de llevar a cabo la minimización del medio.</i>	18
4.1	<i>Composición del flujo de alimentación a la bacteria E. coli. Después de llevar a cabo la minimización del medio.</i>	26
5.1	<i>Diagrama de Gantt del proyecto. Realizado sobre una plantilla de Smarsheet.</i>	28

Lista de tablas

2.1 <i>Resultados de productividad para cada V_{max} estudiado.</i>	9
2.2 <i>Resultados de productividad para V_{max} desde 0.5 a 0.6 mM s⁻¹.</i>	10
2.3 <i>Flujos de las reacciones seleccionadas como más influyentes que proporcionan la máxima productividad.</i>	12
3.1 <i>Nomenclatura de las reacciones equivalentes en los dos modelos.</i>	16
3.2 <i>Flujo de las reacciones de intercambio inicialmente y después de minimizar el medio de crecimiento de la bacteria.</i>	20
3.3 <i>Flujos optimizados de entrada para maximizar la productividad de citramalato.</i>	21
5.1 <i>Distribución temporal del proyecto.</i>	27
5.2 <i>Estimación de costes del proyecto</i>	29

Anexo A

Código Python desarrollado

En el siguiente Anexo se incluye el programa desarrollado a lo largo del trabajo para hallar los resultados presentados. El lenguaje de programación utilizado es Python en el entorno interactivo Spyder3. Se han elaborado un total de cuatro archivos distintos.

A.I Programa principal

En el programa principal se llama a las distintas funciones desarrolladas en los otros archivos.

```
from __future__ import print_function
import cobra.test
import numpy as np
import matplotlib.pyplot as plt
from Kineticopt import ecolicit
from Citrareactions import add_citramalate_reaction
from FBAfunctions import *

#CONSTANTS
FC = 6.372 #Cosntant to change flux units from mmol/s to mmol/gDW*h
GMW = 180.156 #Glucose molecular weight (mmg/mmol)
CMW = 148.11 #Citramalate molecular weight (mmg/mmol)

#Calculation of flux rates in the kinetic model to optimize citramalate
#production
ecit = ecolicit(sbmlfile = "../E_coli_Millard2016.xml") #Addition of the
citramalate reaction to the kinetic model

print("Number of species: ", ecit.model.getNumSpecies())
print("Number of reactions: ", ecit.model.getNumReactions())

vmax_low, vmax_up, no_vmaxes = 0.5, 0.6, 20 # Vmaxes to be tested

#Compute growth for glucose concentrations

Vmax_v = np.linspace(vmax_low, vmax_up, no_vmaxes)
productivities = []
max_product=0

#loop to calculate the productivities for the different Vmax
for Vmax in Vmax_v:    ecit.setVmax("CITRA_SYN", value = Vmax)
    rr = ecit.runmodel()
```

```

productivity = ecit.productcalc(rr = rr, time0 = 0, timef = 200*3600,
npoints = 1000) # Growth in h-1
print("Vmax:", Vmax, "Productivity:", productivity, "h-1")
productivities.append(productivity)
if max_product < productivity:
    Vmaxf=Vmax
    max_product = productivity
print('Vmax with the higher productivity: ', Vmaxf)

plt.plot(Vmax_v, productivities)
plt.xlabel('Vmax')
plt.ylabel('Productivity (h-1)')
plt.show()

ecit.setVmax("CITRA_SYN", value = Vmaxf) #setting the Vmax with the higher
productivity
reactions_fluxes = ecit.getreactionrates(rr = rr, time0 = 0, timef =
200*3600, npoints = 1000) #Calculation of the needed reactions fluxes for the
stoichiometric model, at the highest productivity
gluc_cit_rates = ecit.getcitglucrates(rr = rr, time0 = 0, timef = 200*3600,
npoints = 1000) #Calculation of tglucose and citramalate fluxes for the
stoichiometric model, at the highest productivity
gluc_rate = gluc_cit_rates[0] #glucose reaction rate, it's constant
cit_rate = gluc_cit_rates[1] #citramalate flux of highest productivity
print(reactions_fluxes)

#STOICHIOMETRIC MODEL OPTIMIZATION

model=cobra.io.read_sbml_model('MODEL1108160000.xml') #stoichiometric model of
E. coli metabolism

print(f'{len(model.reactions)}reactions')
print(f'{len(model.metabolites)}metabolites')
print(f'{len(model.genes)}genes')

upcitbound = cit_rate*FC*1.25 #upper bound of the reaction flux rate
lowcitbound = cit_rate*FC*0.75 #lower bound of the reaction flux rate

citreac = add_citramalate_reaction(model = model, upbound = upcitbound,
lowbound = lowcitbound) #addition of the citramalate synthesis reaction to
the model
citreac.add_cit_transport_cp(model = model) # addition of the citramalate
transport reaction from cytoplasm to periplasm
citreac.add_cit_transport_pe(model = model) # addition of the citramalate
transport reaction from periplasm to extracellular
citreac.add_cit_removal(model = model) # output of citramalate produced in
the cell

print("Reactions after:")
print(f'{len(model.reactions)}reactions')
print(f'{len(model.metabolites)}metabolites')
print(f'{len(model.genes)}genes')

percentage = percentage_calculation(model = model, bound_low = 0, bound_up =
0.4, no_bound = 20, cit_rate = cit_rate, bounds_ve = reactions_fluxes,
gluc_rate = gluc_rate, FC = FC, CMW = CMW, GMW = GMW)
# Function to calculate the flux range that provides the maximum growth, flux
rates of the studied reactions in the

```

```

# kinetic model will be modified. For each flux rate obtained in the kinetic
model, the optimum percentage of each value
#between which the flow limits can vary to provide maximum productivity will
be calculated.
# Productivity is defined as : productivity = growth* (citramalate
grams/glucose grams)

print('Flux rate range that maximizes growth (%) : ', percentage)

setfluxranges(model = model, boundpercentage = percentage, bounds_ve =
reactions_fluxes, FC = FC)
#Function to set the calcualted flux range percentage, in the previous step,
that provides the maximum growth

fluxrate = citflux_calculation(model = model, citflux_low = 0.1, citflux_up =
11, no_citflux = 30, gluc_rate = gluc_rate, CMW = CMW, FC = FC, GMW = GMW)
print('Citramalate flux that maximizes productivity (mmol/(gDW*h)): ',
fluxrate)
#Function to maximize productivity setting different flux rates of
citramalate synthesis reaction
productivity0 = max_prod_opt(model = model, fluxrate = fluxrate, GMW = GMW,
FC = FC, CMW = CMW, gluc_rate = gluc_rate)
minimal_medium_function(model = model, fluxrate = fluxrate, gluc_rate =
gluc_rate, FC = FC, CMW = CMW, GMW = GMW, productivity0 = productivity0)
minimize_function(model = model, fluxrate = fluxrate, gluc_rate =
gluc_rate, FC = FC, CMW = CMW, GMW = GMW, productivity0 = productivity0, )
anaerobic(model=model, gluc_rate = gluc_rate, FC = FC, CMW = CMW, GMW =
GMW)

```

A.II Optimización cinética.

En este fichero, se lleva a cabo el análisis del modelo cinético. Se definen las funciones necesarias para añadir la síntesis de citramalato al modelo, calcular su velocidad máxima óptima y el flujo de las reacciones más influyentes en su productividad.

```

from __future__ import division, print_function
import roadrunner
import libsbml

class ecolicit:
    """SBML model of E. coli plust reaction for citramalate synthesis"""

    def __init__(self, sbmlfile, Vmax = 4, Km = 0.495, initial_CITRA =
0.0):
        """
        sbmlfile: SBML file with the model
        Vmax: Vmax of the citramalate synthesis reaction (mM/s)
        Km: Km of the citramalate synthesis reaction (mM)
        initial_CITRA: Initial concentration of citramalate (mM)
        """
        reader = libsbml.SBMLReader()
        self.document = reader.readSBMLFromFile(sbmlfile)
        self.model = self.document.getModel()

```

```

# Create reaction for citramalate synthesis
citrasynt = self.model.createReaction()
citrasynt.setId("CITRA_SYN")
citrasynt.setName("CITRA_SYN");

spr1 = citrasynt.createReactant()
spr1.setSpecies("ACCOA")
spr2 = citrasynt.createReactant()
spr2.setSpecies("PYR")
spr3 = citrasynt.createReactant()
spr3.setSpecies("H2O")
spp1 = citrasynt.createProduct()
spp1.setSpecies("COA")
spp2 = citrasynt.createProduct()
spp2.setSpecies("Hin")

# Kinetic law
kl = citrasynt.createKineticLaw()
para = kl.createParameter()
para.setId("Vmax")
para.setValue(Vmax)
para = kl.createParameter()
para.setId("Km")
para.setValue(Km)
math_ast = libsbml.parseL3Formula('( $V_{max} * ACCOA$ ) / ( $ACCOA + Km$ )')
kl.setMath(math_ast)

def writeToFile(self, filename ="E_coli_Millard2016_CITRA2.xml"):
    libsbml.writeSBMLToFile(self.document, filename)

def setVmax(self, reacId, value):
    # Units: mM/s
    reac = self.model.getReaction(reacId)
    lo = reac.getListOfAllElements()
    for el in lo:
        if type(el) is libsbml.Parameter:
            if el.getId() == 'Vmax':
                el.setValue(value)

def runmodel(self):
    return
    roadrunner.RoadRunner(libsbml.writeSBMLToString(self.document))

def productcalc(self, rr, time0, timef, npoints):

    # Run model and obtain steady state
    selection = ["CITRA_SYN", "GROWTH"]
    rr.timeCourseSelections = selection
    result = rr.simulate(time0, timef, npoints)
    growth = result[-1,selection.index("GROWTH")]*3600
    citmmol = result[-1,selection.index('CITRA_SYN')]
    citgram = citmmol*148.11
    glucmmol = self.model.getParameter('FEED').getValue() #mg/s
    glucgram = glucmmol*180.156 #mg/s
    productivity = growth*(citgram/glucgram) #h-1
    return productivity

def getreactionrates(self, rr, time0, timef, npoints):

```

```

# Run model and obtain steady state
print('REACTION FLUXES: ')
reactions_opt = [ ]*12
reacselection = [ "XCH_P", "PIT", "PGI", "PFK", "FBA", "TPI",
"RPE", "MQO", "PTA", "ACS", "ADK", "CYA", "_ACE_OUT" ]
rr.timeCourseSelections = reacselection
result = rr.simulate(time0, timef, npoints)
for reaction_i in reacselection:
    rate = result[-1,reacselection.index(reaction_i)]
    reactions_opt.append(rate)
    print(reaction_i, ': ', rate)
return reactions_opt

def getcitglucrates(self, rr, time0, timef, npoints):
    selection = [ "CITRA_SYN", 'FEED' ]
    rr.timeCourseSelections = selection
    result = rr.simulate(time0, timef, npoints)
    glucrate = result[-1,selection.index('FEED')]
    citrate = result[-1,selection.index('CITRA_SYN')]
    gluc_cit_rates = [glucrate, citrate]
    return gluc_cit_rates

```

A.III Adición de la síntesis de citramalato al modelo estequiométrico

Aquí se muestra el programa para adicionar la reacción de síntesis del citramalato y las reacciones necesarias de transporte, al modelo estequiométrico.

```

from __future__ import print_function
from cobra import Reaction, Metabolite

class add_citramalate_reaction:
    def __init__(self, model, upbound, lowbound):
        CITRA = Metabolite('citra_c', formula='C5H6O5', name='Critamalate
acid', compartment='c')
        model.add_metabolites([CITRA])
        ACCOA = model.metabolites.get_by_id('accoa_c')
        PYR = model.metabolites.get_by_id('pyr_c')
        H2O = model.metabolites.get_by_id('h2o_c')
        COA = model.metabolites.get_by_id('coa_c')
        Hin = model.metabolites.get_by_id('h_c')
        reaction = Reaction('CIT')
        reaction.name = 'Citramalate synthesis'
        reaction.subsystem = 'Cell Envelope Biosynthesis'
        reaction.lower_bound = lowbound
        reaction.upper_bound = upbound
        reaction.add_metabolites({
            ACCOA: -1.0,
            PYR: -1.0,
            H2O: -1.0,
            COA: 1.0,
            Hin: 1.0,
            CITRA: 1.0
        })

```

```

model.add_reactions([reaction])
print('Citramalate reaction : ', reaction.reaction)

def add_cit_transport_cp(self, model):
    CITRA = model.metabolites.get_by_id('citra_c')
    CITRAp = Metabolite('citra_p', formula='C5H6O5', name='Citrimalate
acid periplasm', compartment='p')
    model.add_metabolites([CITRAp])
    reaction = Reaction('CITRAtexcp')
    reaction.name = 'Citramalate transport via diffusion (cytoplasm
to periplasm)'
    reaction.subsystem = 'Cell Envelope Biosynthesis'
    reaction.add_metabolites({
        CITRA: -1.0,
        CITRAp: 1.0

    })
    model.add_reactions([reaction])
    print('Citramalate transport reaction (from cytoplasm to
periplasm) : ', reaction.reaction)

def add_cit_transport_pe(self, model):
    CITRAp = model.metabolites.get_by_id('citra_p')
    CITRAe = Metabolite('citra_e', formula='C5H6O5', name='Citrimalate
acid extracellular', compartment='e')
    model.add_metabolites([CITRAe])

    reaction = Reaction('CITRAtexpe')
    reaction.name = 'Citramalate transport via diffusion (periplasm
to extracellular)'
    reaction.subsystem = 'Cell Envelope Biosynthesis'
    reaction.add_metabolites({
        CITRAp: -1.0,
        CITRAe: 1.0

    })
    model.add_reactions([reaction])
    print('Citramalate transport via diffusion (from periplasm to
extracellular) : ', reaction.reaction)

def add_cit_removal(self, model):
    CITRAe = model.metabolites.get_by_id('citra_e')

    reaction = Reaction('CITRAremoval')
    reaction.name = 'Citramalate removal from the model'
    reaction.subsystem = 'Cell Envelope Biosynthesis'
    reaction.add_metabolites({
        CITRAe: -1.0,
    })
    model.add_reactions([reaction])
    print('Citramalate removal reaction : ', reaction.reaction)

```

A.IV Análisis de balance de flujo

En este fichero se desarrollan las funciones para optimizar las cotas de flujo y el flujo de citramalato, así como para minimizar el medio de cultivo, en el modelo estequiométrico.

```
from __future__ import print_function
import matplotlib.pyplot as plt
import numpy as np
from cobra.medium import minimal_medium

"""SBML model of E. coli plust reaction for citramalate synthesis"""

def percentage_calculation(model, bound_low, bound_up, no_bound,
bounds_ve, cit_rate, gluc_rate, FC, CMW, GMW):
    flux=0
    max_product=0
    boundper_v = np.linspace(bound_low, bound_up, no_bound)
    productivities = [] #vector to store the calculated productivities
    for each percentage
    growths = [] #vector to store the growth for each percentage
    citram = [] #vector to store citramalates fluxes for each percentage
    glucgram = gluc_rate*GMW*FC #glucose feed in mg/gDW*h

    model.reactions.get_by_id('ACtex').lower_bound = -0.1
    model.reactions.get_by_id('ACtex').upper_bound = 0.1
    model.reactions.get_by_id('GLCtex').lower_bound = gluc_rate*FC
    model.reactions.get_by_id('GLCtex').upper_bound = gluc_rate*FC

    reacselections = ["PItex", "PIt2rpp", "PGI", "PFK", "FBA", "TPI",
"RPE", "MDH2", "PTAr", "ACS", "ADK1", "ADNCYC"]
    #reactions and fluxes calculated in the kinetic model, with which the
    productivity is maximised changing Vmax
    for boundper in boundper_v:
        for (reaction_i, bound) in zip(reacselections, bounds_ve):
            flux = bound*FC #flux rate in mmol/(gDW*h)
            upbound = flux*(1+boundper) #upper flux bound
            lowbound = flux*(1-boundper) #lower flux bound
            model.reactions.get_by_id(reaction_i).lower_bound = lowbound
            model.reactions.get_by_id(reaction_i).upper_bound = upbound
            lowcbound = cit_rate*FC*(1-boundper)
            #lower flux bound for citramalate synthesis
            upcbound = cit_rate*FC*(1+boundper)
            #upper flux bound for citramalate synthesis
            model.reactions.get_by_id('CIT').lower_bound = lowcbound
            model.reactions.get_by_id('CIT').upper_bound = upcbound
            solution = model.optimize()
            growth = solution.objective_value
            growths.append(growth)
            citmmol = model.reactions.CIT.flux
            citgram = citmmol*CMW # citramalate synthesis flux rate in
            grams/gDW*h
            citram.append(citgram)
            productivity = growth*(citgram/glucgram) #h-1
            productivities.append(productivity)
```

```

        if max_product < productivity: #function to save the bound
            percentage with the higher productivity
            bounder_opt = boundper
            max_product = productivity
    print(model.reactions.CIT.flux, ' ', model.reactions.GLCtex.flux)
    plt.plot(boundper_v, productivities)
    plt.xlabel('Flux bounds range %')
    plt.ylabel('Productivity')
    plt.show()
    return bounder_opt

def setfluxranges(model, boundpercentage, bounds_ve, FC):
    reacselections = ["PItex", "PIt2rpp", "PGI", "PFK", "FBA", "TPI",
    "RPE", "MDH2", "PTAr", "ACS", "ADK1", "ADNCYC"]
    for (reaction_i, bound) in zip(reacselections, bounds_ve):
        flux = bound*FC #flux rate in mmol/(gDW*h)
        upbound = flux*(1+boundpercentage)
        lowbound = flux*(1-boundpercentage)
        model.reactions.get_by_id(reaction_i).lower_bound = lowbound
        model.reactions.get_by_id(reaction_i).upper_bound = upbound

def citflux_calculation(model, citflux_low, citflux_up, no_citflux,
gluc_rate, GMW, FC, CMW):
    citflux_v = np.linspace(citflux_low, citflux_up, no_citflux) #vector
    with different values of citramalate synthesis flux rates
    glucgram = gluc_rate*GMW*FC #glucose feed in mg/gDW*h
    productivities = []
    growths = []
    max_product = 0
    max_growth = 0
    for fluxcit in citflux_v:
        model.reactions.get_by_id('CIT').lower_bound = fluxcit
        model.reactions.get_by_id('CIT').upper_bound = fluxcit
        solution = model.optimize()
        growth = solution.objective_value
        growths.append(growth)
        citmmol = model.reactions.CIT.flux
        citgram = citmmol*CMW
        productivity = growth*(citgram/glucgram)
        productivities.append(productivity)
        print('Citramalate flux:', "%5f" % model.reactions.CIT.flux,
        'Growth:', growth, 'Productivity:', "%5f" % productivity)
    #function to save the citramalate flux rate with the higher productivity
    if max_product < productivity:
        citflux_opt = fluxcit
        max_product = productivity
    if max_growth < growth: #function to save the citramalate flux
        rate with the higher productivity
        citflux_opt2 = fluxcit
        max_growth = growth
    plt.plot(citflux_v, growths)
    plt.xlabel('Citramalate flux(mmol/(gDW*h))')
    plt.ylabel('Growth (h-1)')
    plt.show()
    plt.plot(citflux_v, productivities)
    plt.xlabel('Citramalate flux(mmol/(gDW*h))')
    plt.ylabel('Productivity')
    plt.show()
    print ('Citramalate flux that maximizes the growth
    (mmol/(gDW*h)):',citflux_opt2)

```

```

    return citflux_opt

def max_prod_opt( model, fluxrate, GMW, FC, CMW, gluc_rate):
    model.reactions.get_by_id('CIT').lower_bound = fluxrate
    model.reactions.get_by_id('CIT').upper_bound = fluxrate
    ex_reac = ['EX_cb11_LPAREN_e_RPAREN_',
               'EX_fe3_LPAREN_e_RPAREN_',
               'EX_nal_LPAREN_e_RPAREN_',
               'EX_sel_LPAREN_e_RPAREN_',
               'EX_slnt_LPAREN_e_RPAREN_',
               'EX_tungs_LPAREN_e_RPAREN_']
    for exch_reactions in ex_reac:
        model.reactions.get_by_id(exch_reactions).lower_bound = 0
        model.reactions.get_by_id(exch_reactions).upper_bound = 0
    solution = model.optimize()
    growth = solution.objective_value
    glucgram = gluc_rate*GMW*FC
    citgram = fluxrate*CMW
    productivity0 = growth*(citgram/glucgram)
    print (productivity0)

    #    print(model.medium)
    #    print(model.exchanges[0:5])
    in_reacflux=[]
    # exchange reactions obtained with model.medium
    exchangereactions = [ "EX_ca2_LPAREN_e_RPAREN_",
                           "EX_cb11_LPAREN_e_RPAREN_",
                           "EX_co2_LPAREN_e_RPAREN_",
                           "EX_cl_LPAREN_e_RPAREN_",
                           "EX_cobalt2_LPAREN_e_RPAREN_",
                           "EX_cu2_LPAREN_e_RPAREN_",
                           "EX_fe2_LPAREN_e_RPAREN_",
                           "EX_fe3_LPAREN_e_RPAREN_",
                           "EX_glc_LPAREN_e_RPAREN_",
                           "EX_h_LPAREN_e_RPAREN_",
                           "EX_h2o_LPAREN_e_RPAREN_",
                           "EX_k_LPAREN_e_RPAREN_",
                           "EX_mg2_LPAREN_e_RPAREN_",
                           "EX_mn2_LPAREN_e_RPAREN_",
                           "EX_mobd_LPAREN_e_RPAREN_",
                           "EX_na1_LPAREN_e_RPAREN_",
                           "EX_nh4_LPAREN_e_RPAREN_",
                           "EX_ni2_LPAREN_e_RPAREN_",
                           "EX_o2_LPAREN_e_RPAREN_",
                           "EX_pi_LPAREN_e_RPAREN_",
                           "EX_sel_LPAREN_e_RPAREN_",
                           "EX_slnt_LPAREN_e_RPAREN_",
                           "EX_so4_LPAREN_e_RPAREN_",
                           "EX_tungs_LPAREN_e_RPAREN_",
                           "EX_zn2_LPAREN_e_RPAREN_"]

    for reaction_e in exchangereactions:
        fluxr = model.reactions.get_by_id(reaction_e).flux
        in_reacflux.append(fluxr)
    print('Model slim optimization: ', model.slim_optimize())
    max_growth = model.slim_optimize()
    x = minimal_medium(model, max_growth)
    print('Minimal medium :', x)
    for (reaction_e, initflux) in zip(exchangereactions, in_reacflux):
        print(reaction_e)
        print('Initialflux rate :', "%4f" % initflux, 'Final flux rate',
              "%4f" % model.reactions.get_by_id(reaction_e).flux)
    return productivity0
##    minimal_medium(model, 0.8, minimize_components=8,
##open_exchanges=True)

#Function to minize the medium, calculating wich fluxes can be removed
# from the model without affecting to the productivity
def minimal_medium_function(model, fluxrate, GMW, FC, CMW, gluc_rate,
productivity0):
    remove_compounds = []
    notremov = []
    ex_reac = ['EX_cb11_LPAREN_e_RPAREN_',
               'EX_fe3_LPAREN_e_RPAREN_',
               'EX_nal_LPAREN_e_RPAREN_',
               'EX_sel_LPAREN_e_RPAREN_',
               'EX_slnt_LPAREN_e_RPAREN_',
               'EX_tungs_LPAREN_e_RPAREN_']

```

```

'EX_slnt_LPAREN_e_RPAREN_ ', 'EX_tungs_LPAREN_e_RPAREN_ ',
'EX_cobalt2_LPAREN_e_RPAREN_ ', 'EX_mobd_LPAREN_e_RPAREN_ ']
for exch_reactions in ex_reac:
    model.reactions.get_by_id(exch_reactions).lower_bound = 0
    model.reactions.get_by_id(exch_reactions).upper_bound = 0
    solution2 = model.optimize()
    growth2 = solution2.objective_value
    citgram = fluxrate*CMW
    glucgram = gluc_rate*GMW*FC
    productivity = growth2*(citgram/glucgram)
    diff = productivity0-productivity
    if diff > 0.0001:
        model.reactions.get_by_id(exch_reactions).lower_bound = -1000
        model.reactions.get_by_id(exch_reactions).upper_bound = 1000
        notremov.append(exch_reactions)
    else:
        remove_compounds.append(exch_reactions)
print('Exchanges reactions that can be removed:', remove_compounds)
print('Exchanges reactions with flux 0 that cannot be removed :',
notremov)
print('Productivities before-after:', productivity0, ' - ',
productivity)

def minimize_function(model, fluxrate, GMW, FC, CMW, gluc_rate,
productivity0):
    exchangereactions = [ "EX_ca2_LPAREN_e_RPAREN_",
"EX_cl_LPAREN_e_RPAREN_ ", "EX_cobalt2_LPAREN_e_RPAREN_ ",
"EX_cu2_LPAREN_e_RPAREN_ ", "EX_fe2_LPAREN_e_RPAREN_ ",
"EX_k_LPAREN_e_RPAREN_ ", 'EX_mg2_LPAREN_e_RPAREN_ ',
'EX_mn2_LPAREN_e_RPAREN_ ', 'EX_nh4_LPAREN_e_RPAREN_ ',
'EX_ni2_LPAREN_e_RPAREN_ ', 'EX_o2_LPAREN_e_RPAREN_ ',
'EX_pi_LPAREN_e_RPAREN_ ', 'EX_so4_LPAREN_e_RPAREN_ ',
'EX_zn2_LPAREN_e_RPAREN_ ']

    for exch_reac in exchangereactions:
        model.reactions.get_by_id(exch_reac).lower_bound = 0
        model.reactions.get_by_id(exch_reac).upper_bound = 0
        solution3 = model.optimize()
        growth3 = solution3.objective_value
        citgram = fluxrate*CMW
        glucgram = gluc_rate*GMW*FC
        productivity = growth3*(citgram/glucgram)
        model.reactions.get_by_id(exch_reac).lower_bound = -1000
        model.reactions.get_by_id(exch_reac).upper_bound = 1000
        print(exch_reac, ' ->', productivity)

def anaerobic(model, GMW, FC, CMW, gluc_rate):
    model.reactions.get_by_id('CIT').lower_bound = 0
    model.reactions.get_by_id('CIT').upper_bound = 0
    model.reactions.get_by_id('EX_o2_LPAREN_e_RPAREN_ ').lower_bound = 0
    model.reactions.get_by_id('EX_o2_LPAREN_e_RPAREN_ ').upper_bound = 0
    solution4 = model.optimize()
    growth4 = solution4.objective_value
    fluxrate = model.reactions.CIT.flux
    citgram = fluxrate*CMW
    glucgram = gluc_rate*GMW*FC
    productivity = growth4*(citgram/glucgram)
    print(productivity)

```