



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Instituto de Ciencias Agropecuarias

Licenciatura en Ingeniería en Biotecnología

**Identificación y Validación de un Modelo Metabólico Estructurado para
Oleidesulfovibrio alaskensis implementado en Entornos FLOSS**

T E S I S

Que para obtener el grado de:

Licenciada en Ingeniería en Biotecnología

P R E S E N T A

Rodríguez Ramírez Esperanza Elizabeth

Director de tesis

Dr. Pablo Antonio López Pérez

Comité tutorial

Dr. Paul Misael Garza López

Dra. Josefa Espitia López

Dra. Margarita Islas Pelcastre



Tulancingo de Bravo, Hgo.,



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Agropecuarias
Institute of Agricultural Sciences
Área Académica de Ciencias Agrícolas y Forestales
Academic Area of Agricultural and Forestry Sciences

Tulancingo de Bravo, Hgo., a 27 de julio de 2026
Asunto: Autorización de impresión

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Por este conducto y con fundamento en el Título Cuarto, Capítulo I, Artículo 40 del Reglamento de Titulación, le comunico que el jurado que le fue asignado al pasante de Licenciatura en Ingeniería en Biotecnología, Rodríguez Ramírez Esperanza Elizabeth, quien presenta el trabajo de Tesis denominado **"Identificación y Validación de un Modelo Metabólico Estructurado para *Oleidesulfobvibrio alaskensis* implementado en Entornos FLOSS"**, que después de revisarlo en reunión de comité de tesis, ha decidido autorizar la impresión de este, hechas las correcciones que fueron acordadas.

A continuación, se anotan las firmas de conformidad de los miembros del comité de tesis:

DIRECTOR	Dr. Pablo Antonio López Pérez
ASESOR	Dr. Paul Misael Garza López
ASESOR	Dra. Josefa Espitia López
ASESOR	Dra. Margarita Islas Pelcastre

Sin otro particular por el momento, me despido de usted.

Atentamente
"Amor, Orden y Progreso"

Mtro. Luis González de la Rosa
Coordinador de Ingeniería en Biotecnología



"Amor, Orden y Progreso"



Av. Universidad No. 133, Col. San Miguel Huatengo, Santiago
Tulantepec, C.P. 43775, Hidalgo, Mexico.
Teléfono: 7717172000 Ext. 42073
profe_5566@uaeh.edu.mx

uaeh.edu.mx

AGRADECIMIENTOS

A la **UAEH**, por darme los recursos para estudiar y por ser el lugar donde conocí a los mejores amigos y mentores.

A mis padres, **Elizabeth y Rigoberto**, y a mis hermanos, **Antonio y Rigoberto**, por estar siempre conmigo y brindarme su apoyo incondicional.

A mi director de tesis, el **Dr. Pablo**, por darme un lugar, un tema, su tiempo, su paciencia y la confianza de trabajar con usted.

A mis asesores, el **Dr. Paul** y la **Dra. Josefa**, por escucharme y guiarme en mi camino académico.

A mi asesora, la **Dra. Margarita**, por creer en mis capacidades, brindarme su apoyo, su confianza y un lugar seguro donde seguir aprendiendo.

Al **Mtro. De la Rosa**, "Mazapán", por apoyarme, escuchar mis ideas y mis metas, y hacerme ver que podía alcanzarlas... o, en su defecto, seguir aprendiendo en el intento.

A la familia Tovar Díaz: **Khristian, Bertha y Montse**, por apoyarme siempre y hacerme sentir parte de ustedes.

A mis viejos amigos: **Wendy, Marlen, Lupita, Alejandro, Cristopher y Kiran**, por seguir a mi lado y mantener nuestra amistad, aunque la comunicación no sea mi fuerte en estos momentos.

A mis nuevos amigos, **Alexander y Sayde**, por estar conmigo y apoyarme de corazón en esta etapa de mi vida, con tan poco tiempo de conocerme.

A mis gatitos, por desvelarse conmigo cada noche.

DEDICATORIA

A mi novio, **Eduardo Daniel Tovar Díaz**, por estar a mi lado en los buenos y malos momentos. Por tu apoyo y compañía incondicional. Por confiar y creer en mí en cada cosa que hice. Por no dejarme sola, ni permitirme caer o rendirme. Por hacerme mejor persona y ayudarme a superar miedos y heridas del pasado que no causaste. Por tu paciencia infinita y por traer paz y tranquilidad a mi vida. Siempre estaré eternamente agradecida de que hayas estado presente en esta etapa, de que confiaras en que lo lograría, y de que me ayudaras a tener más confianza en mí, en mis sueños y aspiraciones. Que este sea el eterno recordatorio de tu bondad y tu amor sincero hacia mí.

Índice

Glosario.....	10
Resumen	12
Abstract.....	13
Introducción.....	14
1 Capítulo I	16
1.1 Desulfovibrio	17
1.1.1 Taxonomía	17
1.1.2 Propiedades del microorganismo.....	18
1.1.3 Ruta metabólica Desulfovibrio	18
1.1.4 Metabolismo de <i>Oleidesulfovibrio alaskensis</i>	20
1.1.5 Aplicaciones en el ambiente	20
1.2 Modelado de Bioprocesos.....	21
1.2.1 Modelos estructurados	21
1.2.2 Modelos no estructurados	22
1.2.3 Caja negra y caja blanca	22
1.2.4 Ejemplos de modelos basados en la ruta metabólica y cinéticas de reacción.	24
1.2.5 Principios de la ruta metabólica.....	26
1.2.6 Software para modelar	26
1.2.7 Bases de datos.....	28
1.3 Avances y aplicaciones de <i>Oleidesulfovibrio alaskensis</i>	29
1.4 Cálculo de los Grados de Libertad (GL).....	29
2 Capítulo II.....	31
2.1 Justificación	32
2.2 Problemática.....	33
2.3 Objetivos	34

3	Capítulo III.....	35
3.1	Metodología	36
3.1.1	Reconstrucción de la ruta metabólica y recopilación de Datos Cinéticos	36
3.1.2	Formulación del modelo matemático	37
3.1.3	Implementación computacional y análisis del modelo propuesto	37
3.1.4	Expansión del modelo y su respectivo análisis.....	38
3.1.5	Validación interna.....	39
3.2	Limitaciones del estudio	40
4	Capítulo IV	41
4.1	Modelos cinéticos no estructurados experimentales	42
4.1.1	Modelo matemático no estructurado para operación por lote	46
4.2	Propuesta de ruta metabólica	59
4.2.1	Recopilación de información.....	59
4.2.2	Propuesta de mapa metabólico.	67
4.2.3.	Análisis de grados libertad para la ruta metabólica propuesta	69
4.3	Modelo cinético estructurado de la ruta metabólica propuesta.....	72
4.3.1	Modelo estructurado propuesto para la ruta metabólica para operación por lote....	72
4.3.2	Análisis de GL del modelo estructurado propuesto.....	74
	Conclusión	98
	Recomendaciones	98
	Referencias	99
5	Apéndices	107
5.1	Apéndice A	108
5.2	Apéndice B.....	118
5.3	Apendice C.....	123

5.4	Apéndice D	130
5.5	Apéndice E.....	145

Índice de Figuras

Figura 1. Ventajas y desventajas de modelos de caja negra y blanca.	23
Figura 2. Mapa metabólico de <i>Oleidesulfovibrio alaskensis</i>	24
Figura 3. Aspectos generales de las bases de datos bioinformáticas. <i>Fuente: Petrovsky et al., 2023.</i>	28
Figura 4. Metodología general.	36
Figura 5. <i>Etapas para la reconstrucción de la ruta metabólica y recopilación de datos cinéticos.</i>	36
Figura 6. Etapas para la formulación del modelo matemático.	37
Figura 7. Etapas de la implementación computacional y análisis del modelo propuesto.	38
Figura 8. <i>Pasos a seguir para la expansión del modelo y su respectivo análisis.</i>	38
Figura 9. Aspectos importantes de la validación interna del modelo.	39
Figura 10. <i>Datos experimentales de sulfato, Rstudio</i>	42
Figura 11. Datos experimentales de sulfuro, Rstudio.	43
Figura 12. Datos experimentales de biomasa, Rstudio.	44
Figura 13. Cinética de caja negra obtenida con el paquete pyFOOMB.	49
Figura 14. Mapa de calor de sensibilidades paramétricas del modelado de caja negra.	53
Figura 15. Cinética no estructurada de <i>Oleidesulfovibrio alaskensis</i> con distribuciones de los parámetros estimados con pyFOOMB.	55
Figura 16. Cinética no estructurada de <i>Oleidesulfovibrio alaskensis</i> donde se comparan las simulaciones de los diferentes conjuntos de parámetros con pyFOOMB.	56
Figura 17. Propuesta de mapa metabólico de la vía del sulfato y la vía del lactato.	67
Figura 18. Cinética de <i>Oleidesulfovibrio alaskensis</i> de las concentraciones de lactato, acetato, sulfato. Sulfuro, biomasa y carbono, con enzimas involucradas de la ruta del lactato y sulfato...77	77
Figura 19. Concentraciones de todas las variables dinámicas del modelo estructurado.	78
Figura 20. Concentraciones de las variables dinámicas de la vía del sulfato en el modelo estructurado.	78
Figura 21. Concentraciones de las variables dinámicas de la vía del lactato en el modelo estructurado.	79
Figura 22. Mapa de calor de la sensibilidad del modelo estructurado, variables y parámetros. ..	82

Figura 23. Análisis de correlación entre parámetros del modelo expandido y el producto (H_2S).	85
Figura 24. Ranking de los diez parámetros más relevantes en la predicción de formación de producto.	85
Figura 25. <i>Análisis de las trayectorias dinámicas del producto en relación a los parámetros del modelo expandido.</i>	86
Figura 26. Mapa de calor de sensibilidad normalizada de los parámetros sobre las variables dinámicas del modelo estructurado.	91
Figura 27. Comparación de los coeficientes de determinación de las variables dinámicas del modelo estructurado.....	93
Figura 28. Importancia global de los parámetros más influyentes en el modelo estructurado.....	94
Figura 29. Especificidad paramétrica mediante el coeficiente de variación de las sensibilidades a través de las distintas variables de estado.....	95
Figura A 30. Resultados del código para el análisis de sensibilidad del modelo no estructurado.	113
Figura A 31. Gráfico tipo Cornet plot generados por el muestreo de Monte Carlo.	121
Figura A 32. Resultados del código para el análisis de sensibilidad del modelo estructurado.	132

Índice de Tablas

Tabla 1. Ventajas y desventajas de los modelos estructurados.....	22
Tabla 2. Ventajas y limitaciones de modelos no estructurados.	22
Tabla 3. Información de las enzimas disponibles en la base de datos de KEGG de la ruta desasimilatoria de <i>Oleidesulfovibrio alaskensis</i>	25
Tabla 4. Modelos cinéticos no estructurados.....	26
Tabla 5. Clasificación de resultados de GL.	30
Tabla 6. Parámetros cinéticos reportados en la literatura para el diseño del modelado cinético..	45
Tabla 7. Variables del modelo no estructurado.	48
Tabla 8. Sensibilidad de los parámetros.	52
Tabla 9. Resumen de los resultados de las distribuciones marginales, correlaciones entre parámetros, y coeficientes de correlación final.	57
Tabla 10. Reacción bioquímica de la ruta de lactato a acetato mediante bases de datos.	59
Tabla 11. Reacción bioquímica de la ruta de sulfato a sulfuro mediante bases de datos.	60
Tabla 12. Reacción bioquímica de conversión de acetato a sulfuro descrito en KEGG.	61
Tabla 13. Descripción de la vía de sulfato a sulfuro.....	62
Tabla 14. Descripción de la vía de lactato a acetato.....	64
Tabla 15. Complejos transmembrana conservados (TMC) identificados en las rutas metabólicas del lactato y sulfato.....	66
Tabla 16. Variables del modelo estructurado.	74
Tabla 17. Sensibilidad de los parámetros y las distintas variables.	80
Tabla 18. Comparación de las correlaciones de importancia para la concentración de la producción de H ₂ S y la trayectoria.	87
Tabla 19. Valores de R ² , MSE, parámetros importantes y el parámetro más influyente con respecto a las variables respuesta del modelo expandido.	96

Glosario

AprAB: complejo enzimático de la adenosina-5'-fosfosulfato (APS) reductasa.

APS reductasa: La enzima adenosil fosfosulfato reductasa es crucial en la vía respiratoria, trabaja en conjunto al complejo proteico de membrana Qmo para reducir el APS a sulfito.

Asimilatoria: proceso metabólico en microorganismos y plantas donde compuestos inorgánicos se reducen para incorporarse en moléculas esenciales, como proteínas y ácidos nucleicos.

BSR: Bacterias sulfato-reductoras.

Corner plot: es una visualización de datos utilizada para mostrar las relaciones, distribuciones y correlaciones entre múltiples variables de un conjunto de datos multifuncional.

Desasimilatoria: proceso metabólico anaerobio donde microorganismos utilizan compuestos inorgánicos como aceptores finales de electrones en lugar de oxígeno, convirtiéndolos en productos reducidos sin incorporarlos a su biomasa.

Dsr: (generalmente conocido como DsrMKJOP) es un complejo multiproteico integral de membrana fundamental en el metabolismo disimilatorio del azufre, presente tanto en bacterias reductoras de sulfato como en bacterias reductoras de azufre. Actúa como un módulo de transporte de electrones, conectando los procesos redox citoplasmáticos con la cadena respiratoria en la membrana.

EDOs: Ecuaciones diferenciales ordinarias.

FLOSS: es el acrónimo de Free/Libre Open Source Software (Software Libre y de Código Abierto).

Ldh: enzima citoplasmática soluble, en ciertos contextos metabólicos y bacterianos, existen formas de Ldh asociadas a membranas.

Machine-learning: (aprendizaje automático) es una rama de la inteligencia artificial que permite a los sistemas aprender de los datos, identificar patrones y toma de decisiones autónomas sin ser programados explícitamente.

Metabolismo de desbordamiento: (overflow metabolism) se prioriza un crecimiento rápido sobre la eficiencia energética.

MQ: Menaquinona, es la forma oxidada de la quinona (aceptor de electrones).

MQH₂: Menaquinol, es la forma reducida de la quinona (donador de electrones).

MSE: Mean Squared Error, traducido al español como Error Cuadrático Medio, es una métrica utilizada en Machine-learning para evaluar la precisión de un modelo de regresión o estimador.

PCA: Análisis de componentes principales.

PFOR: significa Piruvato:Ferredoxina Oxidorreductasa, cataliza la conversión de piruvato en acetil-CoA y CO₂.

Pool de quinonas: es el nodo central y móvil de transporte de electrones en una membrana.

pyFOOMB: Python Framework for Object-Oriented Modeling of Bioprocesses (Marco de trabajo de Python para el modelado orientados a objetos de bioprocesos).

Qmo: (Quinone-interacting membrane-bound oxidoreductase) es un sistema proteico fundamental en bacterias sulfato reductoras encargado de acoplar la oxidación de menaquinol en la membrana citoplasmática en la reducción de sulfato.

Qrc: (complejo quinona-reductasa, codificado por el operón qrcABCD) complejo proteica transmembrana esencial presente en ciertas bacterias sulfato reductoras. Actúa como un componente clave en la cadena respiratoria anaerobia.

rhs: Right-Hand Side (lado derecho de las ecuaciones diferenciales). Es un término estándar en modelado matemático y sistemas dinámicos. Define como cambian las variables de estado con respecto al tiempo. Es un operador general que define el campo vectorial de un sistema de ecuaciones diferenciales ordinarias (EDO).

Trade-offs: situación en la que se debe elegir entre dos objetivos en conflicto, sacrificando parte de uno para obtener más de otro.

Resumen

Las bacterias sulfato reductoras del género *Oleidesulfovibrio* poseen un gran potencial biotecnológico para la biorremediación de metales pesados, sin embargo, la comprensión cuantitativa de su metabolismo requiere enfoques integrados que combinen el modelado matemático, datos cinéticos e información metabólica detallada. En este trabajo se desarrolló e implementó un modelo cinético estructurado para describir el crecimiento de *Oleidesulfovibrio alaskensis*, integrando las rutas metabólicas de oxidación de lactato y reducción desasimilatoria del sulfato. Además, a partir de datos experimentales reportados en la literatura y mediante la reconstrucción de la ruta metabólica utilizando bases bioinformáticas (*KEGG*, *UniProt*, *BRENDA*), se formularon modelos basados en ecuaciones diferenciales ordinarias (EDOs) que fueron implementados en el entorno *Python* mediante *Jupyter Notebook* utilizando el paquete *pyFOOMB*. Se desarrollaron dos aproximaciones: un modelo no estructurado de seis variables de estado (sulfato, sulfuro, lactato, acetato, biomasa y CO_2) y un modelo estructurado que incorporó metabolitos intermediarios (APS, sulfito, piruvato, acetyl-CoA, acetilfosfato) con sus respectivas cinéticas enzimáticas. Mediante análisis de sensibilidad paramétrica, estimación de Monte Carlo y técnicas de aprendizaje automático, se identificaron los parámetros de mayor influencia en la dinámica del sistema. Los resultados mostraron que parámetros asociados a la inhibición por producto ($SI / \text{H}_2\text{S}$), la tasa máxima de crecimiento (μ_{max}) y los rendimientos (Y , YP) ejercen el mayor control sobre las variables del sistema, con coeficiente de determinación superiores a 0.88 para la predicción de sulfuro. El análisis de correlaciones reveló acoplamientos metabólicos significativos, como la fuerte correlación positiva entre los rendimientos de sustrato y producto ($\rho=0.971$) y las correlaciones negativas entre la velocidad de crecimiento y eficiencia de conversión ($\rho=-0.801$). El modelo estructurado redujo adecuadamente las dinámicas esperadas (R^2 entre 0.80 y 0.99) y permitió identificar tres limitaciones metabólicas principales: inhibición por acumulación de sulfuro, dependencia crítica a la disponibilidad de lactato y restricciones en la conservación de la energía para *O. alaskensis* utilizando herramientas de código abierto, sentando bases para futuros estudios de optimización de bioprocesos.

Palabras clave: *Oleidesulfovibrio alaskensis*, modelado cinético estructurado, bacterias sulfato-reductoras, análisis de sensibilidad, *pyFOOMB*, ruta metabólica.

Abstract

Sulfate-reducing bacteria of the genus *Oleidesulfovibrio* possess great biotechnological potential for the bioremediation of heavy metals; however, a quantitative understanding of their metabolism requires integrated approaches that combine mathematical models with detailed metabolic information. In this work, a structured kinetic model was developed and implemented to describe the growth of *Oleidesulfovibrio alaskensis* 6SR, integrating the metabolic pathways of lactate oxidation and dissimilatory sulfate reduction. Based on experimental data reported in the literature and through the reconstruction of the metabolic pathway using bioinformatics databases (KEGG, UniProt, BRENDA), ordinary differential equation (ODE) models were formulated and implemented in the Python environment using Jupyter Notebook with the pyFOOMB package. Two approaches were developed: an unstructured model with six state variables (sulfate, sulfide, lactate, acetate, biomass, and CO₂) and an expanded model incorporating intermediate metabolites (APS, sulfite, pyruvate, acetyl-CoA, acetylphosphate) with their respective enzyme kinetics. Parametric sensitivity analysis, Monte Carlo estimation, and machine learning techniques were used to identify the parameters with the greatest influence on system dynamics. The results showed that parameters associated with product inhibition (*SI*), maximum growth rate (μ_{max}), and yields (*Y*, *YP*) exert the greatest control over the system variables, with coefficients of determination greater than 0.88 for sulfide prediction. Correlation analysis revealed significant metabolic couplings, such as the strong positive correlation between substrate and product yields ($\rho = 0.971$) and the negative correlations between growth rate and conversion efficiency ($\rho = -0.801$). The structured model adequately reduced the expected dynamics (R^2 between 0.80 and 0.99) and allowed the identification of three main metabolic constraints: inhibition by sulfide accumulation, critical dependence on lactate availability, and energy conservation constraints. This work represents the first structured kinetic modeling approach for *O. alaskensis* using open-source tools, laying the groundwork for future bioprocess optimization studies.

Keywords: *Oleidesulfovibrio alaskensis*, structured kinetic modeling, sulfate-reducing bacteria, sensitivity analysis, pyFOOMB, metabolic pathway.

Introducción

La creciente demanda de soluciones biotecnológicas para la remediación de ambientes contaminados ha posicionado a los microorganismos sulfato-reductores como agentes clave en procesos de biorremediación de metales pesados. Entre estos, el género *Desulfovibrio* ha sido ampliamente estudiado por su capacidad de reducir sulfato a sulfuro, lo que facilita la precipitación y remoción de metales en ambientes acuosos (Álvarez-Márquez et al., 2022). La reclasificación taxonómica reciente ha dado lugar al género *Oleidesulfovibrio*, donde se ubica la especie *Oleidesulfovibrio alaskensis* (anteriormente *Desulfovibrio alaskensis*), una bacteria con la capacidad para tolerar y remover metales pesados mediante la precipitación de sulfuros metálicos (López Pérez, 2013; Sokolova et al., 2021).

El metabolismo de *O. alaskensis* implica dos rutas centrales interconectadas: la oxidación incompleta de lactato a acetato y CO_2 , y la reducción desasimilatoria del sulfato a sulfuro de hidrógeno (Keller et al., 2014), esta última es un proceso multienzimático que requiere la activación del sulfato mediante ATP sulfurilasa para formar adenosina-5-fosfosulfato (APS), seguida de la reducción a sulfito por la APS reductasa y finalmente a sulfuro por el complejo DsrMKJOP y el sulfito reductasa (Pereira et al., 2011; Barbosa et al., 2024). La complejidad en estas rutas, junto con la presencia de mecanismos de inhibición por producto (especialmente por H_2S) y la existencia de múltiples complejos transmembrana involucrados en el flujo de electrones (QmoABC, DsrMKJOP, QrcABCD), hace que el comportamiento dinámico de *O. alaskensis* en cultivo sea difícil de predecir únicamente con aproximaciones experimentales (Keller et al., 2014).

En este contexto, el modelado matemático de bioprocesos surge como una herramienta para describir cuantitativamente el crecimiento microbiano, el consumo de sustratos y la formación de productos. Las estructuras cinéticas pueden clasificarse en no estructurados, que consideran la biomasa como una entidad homogénea (como los son los modelos basados en Monod), y estructurados, que incorporan información de rutas metabólicas y reacciones enzimáticas específicas, permitiendo una representación más detallada de la fisiología celular. Adicionalmente, los modelos pueden ser de “caja negra” (basados en relaciones de entrada-salida) o de “caja blanca” (fundamentados en mecanismos bioquímicos) (Garibay-Benitez, 2017).

A pesar de los avances de la caracterización genómica y fisiológica de *Oleidesulfovibrio alaskensis*, los trabajos que integran su metabolismo en modelos matemáticos dinámicos son muy

pocos. Los estudios existentes se han centrado principalmente en modelos no estructurados para describir el crecimiento y la remoción de metales (López-Pérez, 2013), sin explorar en profundidad la influencia de los intermediarios metabólicos, y análisis sistemáticos de sensibilidad paramétrica. Esta limitación dificulta la comprensión de los factores que controlan la dinámica del sistema y la identificación de parámetros clave para el diseño y optimización de bioprocesos.

El desarrollo de herramientas computacionales de código abierto (FLOSS) como pyFOOMB ha democratizado el acceso al modelado dinámico, permitiendo a usuarios de conocimiento básicos en programación implementar, simular y analizar modelos simples y complejos de manera reproducible (Hemmerich et al., 2021). La combinación de pyFOOMB con bibliotecas de análisis de datos (*scikit-learn*, *NumPy*, *Matplotlib*) y entornos como Jupyter Notebook facilita la integración de técnicas para la selección de parámetros y el análisis de incertidumbre, abriendo nuevas posibilidades para el estudio de sistemas microbianos.

Por lo tanto, en este estudio desarrollo e implemento un modelo cinético estructurado para *Oleidesulfobrevibacterium alaskensis* que integra las rutas metabólicas de oxidación de lactato y reducción de sulfato utilizando herramientas de código abierto. Además, utilizando datos experimentales e información de parámetros cinéticos obtenidas de bases de datos bioinformáticas como *KEGG*, *UniProt* y *BRENDA*. Una de las aportaciones es la propuesta de la ruta metabólica basada principalmente en el metabolismo extra e intra celular para la formación de sulfuro de hidrogeno y la ruta de degradación de lactato. La propuesta se fundamenta en la información de las bases bioinformáticas, lo que permitió formular los balances de materia y energía descritas por ecuaciones diferenciales ordinarias para cada paso en la ruta metabólica (cinética) que se implementan utilizando *pyFOOMB* en *python*. Mediante análisis de sensibilidad paramétrica, estimación de incertidumbre por Monte Carlo, se identificaron los parámetros de mayor influencia en la dinámica del sistema propuesto. Además, se evaluó la capacidad predictiva del modelo mediante un análisis de GL. Los resultados permitieron comprender mejor las limitaciones metabólicas del género *O. alaskensis* y sentar las bases para futuras aplicaciones en biorremediación y optimización de procesos.

Capítulo I

Desulfovibrio

Modelado de bioprocesos

Principios de ruta metabólica

Avances y aplicaciones de *Oleidesulfovibrio alaskensis*

1.1 Desulfovibrio

Neria-González (s.f.), mencionaron que el género *Desulfovibrio* entre las bacterias reductoras de sulfato, son nutricionalmente similares al crecer en medios de cultivo descritos por Postgate denominados B, C, D y E, enriquecidos por lactato, CO₂, acetato o extracto de levadura como fuente de carbono. Debido a su metabolismo anaerobio son capaces de utilizar el sulfato como aceptor final de electrones, esto permite obtener energía a través de la oxidación de compuestos orgánicos, como el hidrógeno o el formiato. Se caracterizan por tener forma de bacilos curvos (vibro), pero también puede haber en forma de espiroidales y bacilos rectos. Su morfología está sujeta a la etapa de crecimiento del cultivo y los factores ambientales en los que se desarrollan, los cultivos jóvenes presentan formas constantes, pero en cultivos más viejos o estresados sus formas son variadas. La mayoría oxidan parcialmente lactato a acetato y otros de manera completa al pasar del acetato a CO₂, usando el H₂ como fuente de energía. Se consideran halófilos moderados (25 – 35 °C, límite en 44°C), algunos son termofílicos y se conoce que hay otros capaces de crecer bajo 0°C. Son bien conocidos por albergar en su interior Desulfovibrindina-sulfato-reductasa (DSR) y tetramero de citocromo C₃, característicos del metabolismo sulfato reductor. Pueden recurrir a otros compuestos sulfurados como aceptor final de e⁻ (como el sulfito o el tiosulfato) reduciéndolos a H₂S, pero en algunos casos pueden llegar a reducir nitratos. El interés biotecnológico de este género radica en su habilidad de reducir uranio (VI), cromo (VI) y hierro (III).

1.1.1 Taxonomía

Oleidesulfovibrio alaskensis pertenece a la familia *Desulfovibrionaceae*. Dentro de esta familia, de acuerdo con Sinh, Gnimpieba y Sani (2023), recientemente se ha propuesto la creación de dos nuevos géneros: *Alkalidesulfovibrio* y *Desulfovibrionia*.

Sin embargo, la clasificación taxonómica actualizada del NCBI y Raya et al. (2026) la especie *Oleidesulfovibrio alaskensis* es:

- Dominio: Bacteria
- Reino: Pseudomonadati
- Filo: Thermodesulfobacteriota
- Clase: Desulfovibrionia
- Orden: Desulfovibrionales
- Familia: *Desulfovibrionaceae*

- Género: *Oleidesulfovibrio*

1.1.2 Propiedades del microorganismo

Las bacterias del género de *Desulfovibrio* son anaerobias estrictas gram (-) que pueden ser encontradas en el suelo, el agua e inclusive en el tracto intestinal animal o humano. Hoy en día se conoce que tienen forma de bacilos curvos (también es denominada forma de vibrio), pero también se han encontrado bacilos rectos e incluso espiroidales. Estos microorganismos pueden moverse gracias a que cuentan con flagelos polares. La forma de esta especie varía de acuerdo a la edad del mismo, entre más longevos son los cultivos, más células polimórficas alberga, aunque otra causa del polimorfismo es el estrés. Entre las formas que podemos encontrar son sigmoideo, bacilos alargados y curvos, bacilos regordetes o cocoides. Los cultivos se alimentan a partir de lactato, piruvato o CO₂-hidrógeno, necesitan de un agente reductor y dependiendo la especie requieren de vitaminas (Neria-González, s.f.).

O. alaskensis es reconocida por su alta resistencia metálica al tener mecanismos que le permiten abatir iones metálicos en la remoción y recuperación de metales pesados de ambientes contaminados, pueden crecer óptimamente en medios con 30% de NaCl, un pH de 7, a temperatura de 45°C mientras producen EPS, este contiene ramnosa, galactosa, glucosa y arabinosa, así mismo, posee propiedades quelantes facilita la absorción de metales pesados. Su biofilm es lo que les permite protegerse del estrés ambiental y sustancias tóxicas de sus alrededores. Se ha reportado que *O. alaskensis* es capaz de tolerar 18 mg/L de Cr⁺⁶, 200 mg/L de Cd⁺², 130 mg/L de Zn⁺² y 200 mg/L de Pb⁺², adquiridos de ambientes acuosos desde un 78% hasta 98% en forma de sulfuros metálicos, a través de la adsorción en la producción del EPS y en la acumulación metálica en espacios periplasmáticos de la bacteria. Por ello, los mecanismos de remoción de metales pesados se relacionan a la producción de H₂S obtenida de la respiración anaerobia y la resistencia al cromo por parte de la bacteria, puede relacionarse a la ausencia de la enzima cromo-reductasa.

1.1.3 Ruta metabólica *Desulfovibrio*

De acuerdo con Keller et al., (2014), es común que el género de *Desulfovibrio* utilicen como donantes de electrones a sustratos de ácidos orgánicos, H₂, alcoholes de cadena corta o formiato para la reducción del sulfato, este es un proceso que se lleva a cabo en el citoplasma y es mediado por enzimas solubles. Se conoce que cuando se usa H₂ como flujo de electrones debe de haber una presión ambiental de 10 Pa; que la primera reducción de sulfato a sulfito se activa con dos

equivalentes de ATP por acción de la adeniltransferasa, posteriormente se produce APS o también llamada adenosina fosfosulfato aumentando el potencial redox que garantiza la reducción por electrones de hidrógeno, así el APS puede ser reducido por la acción de APS reductasa para obtener sulfito y AMP (adenosinmonofosfato). Por último, se da la reducción de seis electrones de sulfito a sulfuro por la acción del sulfito reductasa desasimilatoria, esta última se encuentra aparentemente ligada al flujo de transporte de electrones.

Keller et al., (2014) confirma que las cepas de *Desulfovibrio* oxidan de manera incompleta sustratos como el piruvato y el lactato. La enzima utilizada para este fin es el lactato deshidrogenasa independiente del nucleótido de pirimidina ligada a la membrana que mira al citoplasma, transforma el lactato en piruvato; posteriormente la oxidación del piruvato en acetil CoA y CO₂ se da gracias al piruvato ferredoxina oxidoreductasa soluble.

Así mismo, Keller et al., (2014, como se cita en Postgate, 1984) dice que la especie de *Desulfovibrio*, además del crecimiento por la reducción del sulfato, también puede fermentar piruvato o lactato con fosforilación a nivel de sustrato conservando la energía de la Acetil CoA. Keller (2014, como se cita en Pereira, 2007) afirma que, para que se lleve a cabo la reducción de Sulfato con electrones de H₂, es necesario la oxidación del hidrógeno en el periplasma por la acción de múltiples isoenzimas de la hidrogenasa y los electrones derivados de este proceso, son llevados al citoplasma por la existencia de vías para el flujo de electrones que parten desde el periplasma. Cuando la obtención de electrones es a partir de ácidos orgánicos, la oxidación tiene lugar directamente en el citoplasma. Es importante mencionar que, Keller et al., (2014, como se citó en Odom, 1981), destaca que la energía generada de la oxidación del lactato participa en un mecanismo de ciclado de hidrógeno que también proporciona la energía metabólica disponible para la reducción del sulfato.

Otro aspecto importante, según Keller et al., (2014, previamente citado en Thauer, 2007; Matias, 2005 y Pereira, 2011) es el papel de dos complejos transmembrana conservados (TMC), la QmoABC (esencial para la reducción del sulfato) y la DsrMKJOP, ya que son los responsables de proporcionar las vías necesarias para que los electrones requeridos lleguen a las enzimas APS reductasa y el bisulfito reductasa, respectivamente. Sin embargo, aún no se ha demostrado que la QmoABC tenga establecido el flujo de electrones desde y hacia el TMC.

Por último, Keller et al., (2014 como se citó en Venceslau, 2010; Li, 2009 y Venceslau, 2011) menciona que recientemente se descubrió un tercer TMC en dos géneros de *Desulfovibrio*, denominado QroABCD, que está asociado directamente con la respiración por sulfato, este complejo recibe electrones de deshidrogenasas del periplasma y de las deshidrogenasas de formato por el citocromo c_3 tetrahemo tipo I (abreviada como Tpl c_3).

1.1.4 Metabolismo de *Oleidesulfovibrio alaskensis*

Oleidesulfovibrio alaskensis es una bacteria anaerobia estricta perteneciente a la familia *Desulfovibrionaceae*, cuya principal vía de obtención de energía es la respiración por sulfato, donde utiliza este anión como aceptor final de electrones, y a su vez obtener energía mediante la oxidación de compuestos orgánicos como lactato, piruvato o hidrógeno. La reducción del sulfato ocurre mediante una secuencia de reacciones citoplasmáticas que incluyen la activación del sulfato adenosina-5-fosfosulfato (APS), seguida de la reducción a sulfito y finalmente a sulfuro, estos procesos son catalizados por complejos enzimáticos conservados como QmoABC y DsrMKJOP (Keller et al., 2014). El lactato es uno de los principales donadores de electrones y fuentes de carbono para *O. alaskensis*, siendo oxidado inicialmente a piruvato y posteriormente a acetil-CoA, con generación de acetato como producto final. Esta dualidad funcional convierte al lactato en un metabolito clave cuya disponibilidad condiciona tanto el balance energético como la eficiencia metabólica del sistema. Adicionalmente, también puede fermentar piruvato o lactato mediante fosforilación a nivel de sustrato, conservando la energía de la acetil-CoA (Pereira, 2007). A pesar de su naturaleza sulfato reductora, su desarrollo se ve inhibido en presencia de H_2S (Neria-González et al., s.f.). Esta inhibición por producto constituye un factor crítico en la regulación del crecimiento y la actividad metabólica del microorganismo.

1.1.5 Aplicaciones en el ambiente

Una de las principales problemáticas ambientales que prevalecen hoy en día, es la contaminación por metales pesados como el plomo, cobre, zinc y el cadmio. Ante esta problemática se propone el uso de microorganismo sulfato reductores por su capacidad de eliminar sulfato, metales pesados y radionucleidos, como el uranio, de efluentes de desechos (Vita et. al., 2015). El género *Desulfovibrio* es de los más utilizados para biorremediación por su capacidad de liberar sulfuro de hidrógeno y HCO_3 en presencia de sulfato y iones de hidrógeno. La liberación de H_2S por parte del microorganismo provoca que los metales disueltos en el agua precipitan en forma de sulfuros

metálicos y queden atrapados en el exopolisacárido de la bacteria (Álvarez, 2022). La importancia de *Oleidesulfobivrio alaskensis* radica en su capacidad de aislar metales contaminantes, como el cadmio, del agua ácida (López-Pérez, 2013; López, et al. 2015; Neria-González, et al. 2016).

1.2 Modelado de Bioprocesos

El modelado matemático de bioprocesos permite describir cuantitativamente la dinámica de crecimiento microbiano, el consumo de sustratos y la formación de productos. Los modelos cinéticos no estructurados, como los basados en la ecuación de Monod, ofrecen una aproximación simplificada que relaciona la tasa de crecimiento a la concentración de sustrato limitante; sin embargo, estos modelos no incorporan mecanismos metabólicos internos ni efectos de inhibición complejos (Hemmerich, et al. 2021).

En contraste, los modelos cinéticos estructurados integran información de rutas metabólicas y reacciones bioquímicas específicas, permitiendo identificar cuellos de botella metabólicos y limitaciones energéticas internas (Garibay, 2017), este enfoque resulta particularmente relevante para los microorganismos sulfato-reductores, donde la acumulación de sulfuro y la eficiencia de conservación de energía desempeñan un papel determinante en la dinámica del crecimiento (Peña-Caballero, et al. 2016; Pérez, et al. 2024; López, 2025).

1.2.1 Modelos estructurados

En este tipo de modelos, la célula es vista como un sistema complejo debido a que se consideran elementos claves del proceso como los nutrientes que se necesitan para el crecimiento, la manera en que se inhibe el crecimiento, los metabolitos que se generan, los sustratos intermediarios y las velocidades del proceso. A continuación, se menciona la clasificación de modelos estructurados de acuerdo con Garibay (2017):

- Modelos estructurados simples: considera a la célula como una unidad homogénea y está basado en mecanismos celulares definidos.
- Modelos morfológicamente estructurados: estructura celular y la distribución de sus componentes.
- Químicamente estructurados: implica las composiciones químicas, la concentración de diferentes moléculas y la dinámica de sus reacciones.

- Genéticamente estructurados: información sobre el genoma, la expresión genética y la regulación de la expresión de genes
- Metabólicamente estructurados: considera las rutas metabólicas de la célula y las reacciones químicas que ocurren en ellas.

Tabla 1. *Ventajas y desventajas de los modelos estructurados.*

Ventajas	Limitaciones
Describe la fisiología celular y mecanismos internos	Requieren datos detallados (como la concentración de enzima)
Ayudan a entender limitaciones enzimáticas o regulación génica	Son computacionalmente complejos

Fuente: Garibay (2017)

1.2.2 Modelos no estructurados

Simplifican el comportamiento del metabolismo celular donde se considera a la biomasa como una entidad homogénea, sin detallar sus componentes y limitándose a una expresión matemática, este tipo de modelos a su vez se dividen en dos grupos de acuerdo con Garibay (2017) los cuales son:

- Estocásticos: considera la variación aleatoria de los parámetros respecto al tiempo, la salida no será la misma debido a la variación de parámetros.
- Deterministas: es el único modelo predictivo donde a cada entrada le pertenece una sola salida (Aguilar, et al. 2017; López, et al. 2013; López, et al. 2025).

Tabla 2. *Ventajas y limitaciones de modelos no estructurados.*

Ventajas	Limitaciones
Requieren pocos datos experimentales.	No detallan mecanismos metabólicos.
Ideales para sistemas complejos con sustratos heterogéneos.	Dependencia a datos empíricos.

Fuente: Garibay (2017)

1.2.3 Caja negra y caja blanca

Los modelos de caja negra definen al sistema de estudio sin considerar mecanismos internos, solo relacionan lo que entra al sistema con lo que sale de él mediante funciones matemáticas, estos

están basados en datos observables y tiene aplicaciones comunes la predicción de crecimiento microbiano en biorreactores o en la optimización de procesos industriales (Gualdron, s.f).

Los modelos de caja blanca definen al sistema de estudio con detalles, incorporan mecanismos internos, ya sean bioquímicos, termodinámicos, genéticos, entre otros; y utilizan ecuaciones matemáticas basadas en leyes físicas o biológicas, este tiene aplicaciones típicas en el modelado de rutas metabólicas (Gualdron, s.f.).

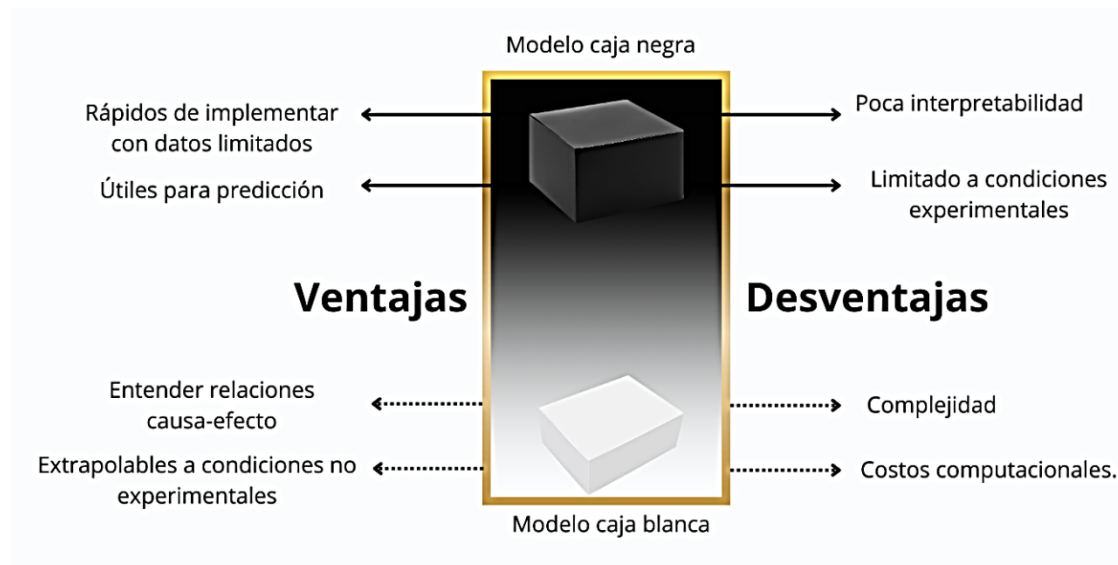


Figura 1. *Ventajas y desventajas de modelos de caja negra y blanca.*

Fuente: elaboración propia.

1.2.4 Ejemplos de modelos basados en la ruta metabólica y cinéticas de reacción.

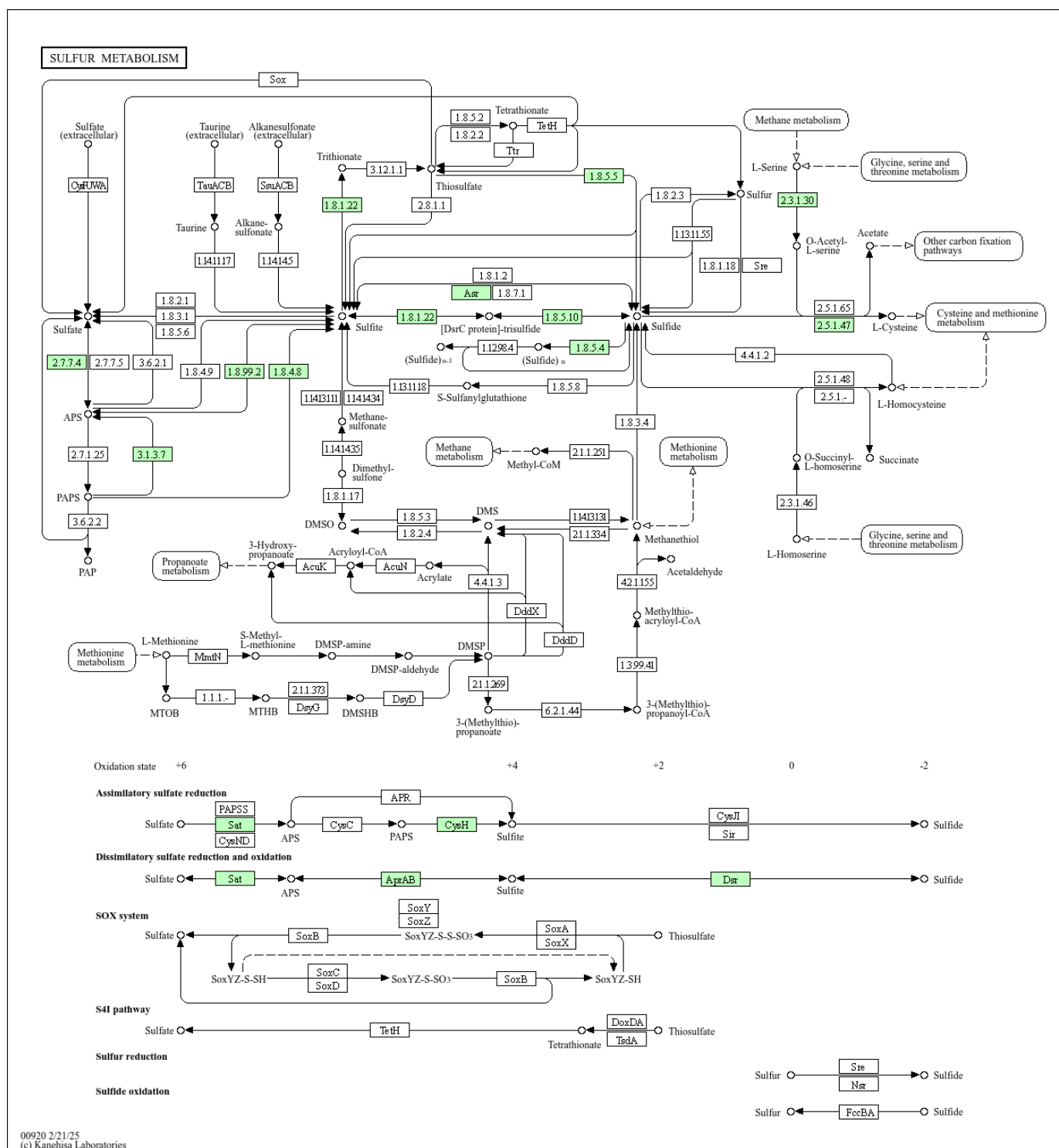


Figura 2. Mapa metabólico de *Oleidesulfovibrio alaskensis*.

Fuente: <https://www.kegg.jp/pathway/dde00920+M00596>

La figura 2 muestra el mapa metabólico de la vía del sulfato descrito para *O. alaskensis* y se encuentra en la base de datos KEGG. En él destacan las enzimas que participan en la reducción desasimilatoria del sulfato en color verde, y de las cuales se obtiene la tabla 3 que se muestra a continuación.

Tabla 3. Información de las enzimas disponibles en la base de datos de KEGG de la ruta desasimilatoria de *Oleidesulfovibrio alaskensis*.

Código (EC)	Abreviatura de enzimas	Nombre completo	Reacción
2.7.7.4 (1)*	ATP-sulfurilasa	Sulfato adenilil transferasa	ATP + sulfato $\rightleftharpoons$ difosfato + sulfato de adenililo
1.8.99.2 (2)*	APS-reductasa	Adenosina 5'-fosfosulfato reductasa	AMP + sulfito + aceptor $\rightleftharpoons$ sulfato de adenililo + aceptor reducido
1.8.1.22 (3)*	DsrABL	Siroheme sulfito reductasa	[proteína DsrC]-trisulfuro + NAD ⁺ + 3 H ₂ O $\rightleftharpoons$ [proteína DsrC]-ditiol + sulfito + NADH + H ⁺
1.8.5.10 (4)*	DsrMKJOP	[DsrC]-trisulfuro reductasa	sulfuro de hidrógeno + un [proteína DsrC]-ditiol + 2 quinonas $\rightleftharpoons$ un [proteína DsrC]-trisulfuro + 2 quinoles
1.8.1.18 (5)*	NADPH NSR S0 reductasa	NAD(P)H azufre oxidorreductasa (dependiente de CoA)	sulfuro de hidrógeno + NAD(P) ⁺ $\rightleftharpoons$ azufre + NAD(P)H + H ⁺
2.5.1.47 (6)*	Cisteína sintasa	O-acetil-L-serina sulfhidrilasa	O-acetil-L-serina + sulfuro de hidrógeno $\rightleftharpoons$ L-cisteína + acetato

Nota: link de la información disponible en KEGG.

- (1) * <https://www.kegg.jp/entry/2.7.7.4>
- (2) * <https://www.kegg.jp/entry/1.8.99.2>
- (3) * <https://www.kegg.jp/entry/1.8.1.22>
- (4) * <https://www.kegg.jp/entry/1.8.5.10>
- (5) * <https://www.kegg.jp/entry/1.8.1.18>
- (6) * <https://www.kegg.jp/entry/2.5.1.47>

1.2.5 Principios de la ruta metabólica

Tabla 4. Modelos cinéticos no estructurados

Modelo cinético por inhibición de producto	Ecuación
Boulton	$\left[\frac{\mu_m k_p}{P + k_p} \right]$
Levenspiel	$\mu_m \left[1 - \frac{P}{k_p} \right]^m$
Luong	$\mu_m \left[1 - \left(\frac{P}{k_p} \right)^m \right]$
Modelos cinéticos modificados	Ecuación
Levenspiel	$\mu(S,P,X) = \mu_m \left[\frac{S}{k_s + S} \right] \left[1 - \left(\frac{P}{P} \right) \right] X$
Haldane-Luong	$\mu(S,P,X) = \left[\frac{\mu_m S}{k_s + S + S^2 k_l^{-1}} \right] \left[1 - \left(\frac{P}{k_p} \right)^m \right] X$
Moser-Luong	$\mu(S,P,X) = \left[\frac{\mu_m S^n}{k_s + S^n} \right] \left[1 - \left(\frac{P}{k_p} \right)^m \right] X$
Aiba	$\mu = \left[\frac{\mu_m S^n}{k_s + S^n} \right] \exp \left(\frac{-S}{k_1} \right)$

Fuente: López-Pérez, (2025).

El modelo matemático para esta investigación se estructura de ecuaciones diferenciales no lineales acopladas basadas en el modelo modificado de Levenspiel sobre respiración bacteriana, el modelo modificado de Moser-Boulton para el consumo de carbono (López-Pérez, 2013).

1.2.6 Software para modelar

1.2.6.1 ¿Qué es anaconda?

Anaconda es un administrador de paquetes con lenguaje de *Python* utilizados generalmente para la ciencia de datos, está relacionada directamente con conceptos de *machine-learning*, contiene herramientas típicas como *Numpy* y *Matplotlib* (Fazt, 2020).

Como desarrolladores de *Python*, se pueden generar proyectos dentro del entorno *Conda*, que a su vez contengan distintos paquetes o dependencias específicas para el proyecto a realizar. Cabe mencionar que por cada proyecto realizado puede requerir de distintas por lo que al trabajar

en un entorno virtual de *Conda*, brindara paquetes de las versiones que se necesitan sin que interfieran entre ellos o en otros proyectos, es decir, permite crear entornos específicos para cada proyecto (Fazt, 2020).

Anaconda es una distribución de *Python*, existen otras distribuciones como lo es *Javthon* (donde *Python* usa el entorno de *Java*), *Cthon* (donde *Python* utiliza el entorno de C). Así que Anaconda usa un programa llamado *Conda*, el cual, permite instalar distintos paquetes y administrar entornos virtuales (Fazt, 2020).

1.2.6.2 *Jupyter notebook*

Existe *Jupyter Notebook* y *Jupyter Lab*, la diferencia entre uno y otro es que, *Jupyter Notebook* permite trabajar un solo cuaderno, mientras que en *Jupyter lab* es para trabajar varios Notebook al mismo tiempo. Sin embargo, en esta ocasión este estudio se enfoca en *Jupyter Notebook*.

La interfaz de *Jupyter Notebook* es una aplicación web que permite crear y compartir códigos de programación. Los *Notebook* tienen la peculiaridad de que utiliza celdas para su desarrollo, esto permite poder visualizar la función del código apenas se termina de escribir una celda, de igual manera permite guardar las imágenes, tablas e incluso gráficas. Por último y no menos importante, permite compartir los códigos de manera fácil y rápida (González, 2020).

Entre las cosas más importantes de esta aplicación son las celdas para ejecutar los códigos, ya que también se puede obtener un formato de texto, títulos y fórmulas utilizando esas las mismas celdas al utilizar la opción de “Markdown” y algunos caracteres especiales como los “#”. Finalmente, las celdas también permiten incluir imágenes externas a la aplicación en la opción para el formato de texto.

1.2.6.3 *pyFOOMB*

El paquete *pyFOOMB* permite la implementación de modelos para bioprocesos de manera flexible y guiada mediante sistemas de ecuaciones diferenciales ordinarias (EDO), se basa en el lenguaje de programación de *Python*, implementa modelos orientados a objetos de bioprocesos de manera rápida y flexible, permite la formulación, la validación y la aplicación de modelos de bioprocesos (Hemmerich, et al. 2021). El análisis de sensibilidad constituye una herramienta fundamental para evaluar la influencia de los parámetros cinéticos sobre las variables de estado del modelo, mediante

este análisis es posible identificar parámetros dominantes, evaluar incertidumbre y detectar correlaciones que reflejan posibles trade-offs metabólicos (Saltelli et al., 2008).

1.2.6.4 Scikit-learn

Es una biblioteca de *Python* que proporciona herramientas simples y eficientes para el análisis predictivo de datos que implementa algoritmos comunes de *machine-learning*, entre sus funciones principales se encuentra la clasificación, regresión, clustering, procesamiento de datos, la selección de modelos y su evaluación (Pedregosa, et al. 2011).

1.2.6.5 RStudio

Es un entorno de desarrollo integrado, gratuito y de código abierto diseñado para el análisis estadístico, manipulación de datos y visualización de datos mediante el lenguaje de programación R (Fazt, 2020).

1.2.7 Bases de datos

Es un aspecto fundamental en la recopilación de la información, ya que las bases de datos garantizan la fidelidad de la información al estar respaldada con textos científicos. Así mismo, el renombre de la base de datos habla de la confiabilidad de su contenido. La figura 3 muestra las bases de datos utilizadas para la elaboración de este documento (Petrovsky et al., 2023).

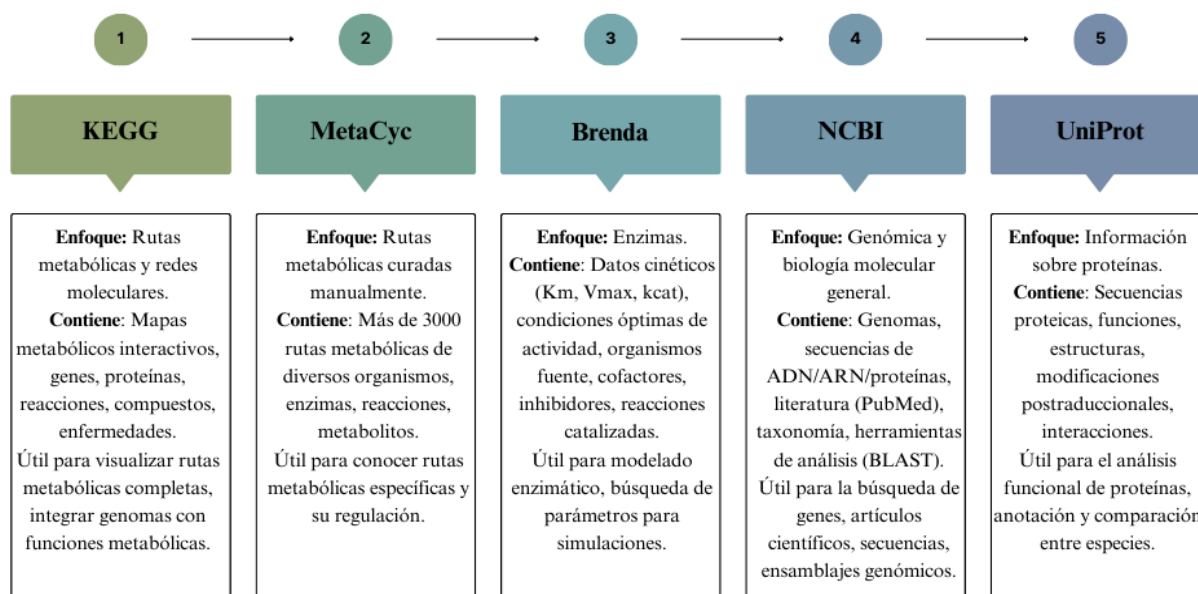


Figura 3. Aspectos generales de las bases de datos bioinformáticas. *Fuente: Petrovsky et al., 2023.*

Nota: describe el principal uso de la base de datos y usos principales.

1.3 Avances y aplicaciones de *Oleidesulfovibrio alaskensis*

De acuerdo con Priya, et al (2025) se evidenció la capacidad de *O. alaskensis* para la producción significativa de hidrógeno en condiciones tanto ácidas como alcalinas. En pH alcalinos las bacterias acidifican su citoplasma mediante el bombeo de protones o la síntesis de sustancias ácidas: De igual manera, las células emplean los mecanismos inversos para disminuir la acidificación por pH bajos. Además, encontraron que las ATPasas del microorganismo funcionan como bombas de protones y sus hidrogenasas llevan a cabo la conversión reversible de protones a hidrógeno.

Sokolova D.S. et al (2021) muestra la capacidad de las bacterias sulfato reductoras (SRB) para resistir a biocidas. Donde la cepa de *Desulfovibrio alaskensis* Kaz19, es capaz de resistir agentes biocidas como el glutaraldehído y el Rancid. Keskin (2021) habla del potencial biorremediador de *O. alaskensis* para corroer metales como el cobre mediante la formación de su biofilm.

Recientemente se ha descubierto que *O. alaskensis* tiene aplicaciones en la industria del biogás debido a que el hidrógeno induce a la dispersión y desprendimientos de biofilms subterráneos en presencia de este microorganismo, disminuyendo los riesgos de bioobstrucción de poros durante los procesos de almacenamiento de hidrógeno (Liu, Ostertag-Henning, Fernø, & Dopffel, 2025). Finalmente, de acuerdo con Isokpehi et. al. (2021) otras especies de *Desulfovibrio* codifican proteínas de estrés universal, lo que les confiere mecanismos de adaptación a condiciones ambientales adversas.

1.4 Cálculo de los Grados de Libertad (GL)

Los grados de libertad determinan si el sistema matemático tiene solución, si faltan datos o si está sobre-especificado. La ecuación matemática es (Coria, s.f.):

$$GL = F - N - M$$

Aquí,

F: es el número de flujos no medidos (incógnitas),

M: es el número de flujos medidos directamente,

N: es el número de nodos independientes (ecuaciones de balance).

- **Flujos (J o v_i):** Son las velocidades o tasas a las que ocurren las reacciones químicas internas y los transportes a través de la membrana. Se expresan típicamente en unidades de masa o cantidad de sustancia por unidad de tiempo y biomasa.
- **Nodos (Metabolitos Internos):** Son los compuestos químicos intermedios que se encuentran dentro del citoplasma o periplasma y que están sujetos al principio de conservación de la masa.
- **Estado Estacionario (Steady State):** Es el supuesto metabólico clave que establece que no hay acumulación de metabolitos internos en el tiempo. Por lo tanto, para cada nodo, la suma de los flujos que lo producen debe ser igual a la suma de los flujos que lo consumen

$$\frac{dC_i}{dt} = 0$$
- **Mediciones Extracelulares:** Son los flujos de consumo de sustratos o producción de metabolitos/gases que cruzan la membrana celular y pueden cuantificarse directamente en el biorreactor (medio de cultivo).

Una vez aplicado el cálculo numérico, el resultado de los GL clasificará el sistema en uno de los siguientes tres escenarios analíticos (tabla 5) (Coria, s.f.):

Tabla 5. *Clasificación de resultados de GL.*

Valor de GL	Clasificación Matemática	Significado Biológico / Práctico
GL>0	Subdeterminado (Indeterminado)	Faltan mediciones. El sistema tiene infinitas soluciones metabólicas posibles. Para resolverlo, se requiere utilizar FBA (Flux Balance Analysis) mediante la optimización de una función objetivo (ej. maximizar biomasa).
GL=0	Determinado	Sistema exacto. Existe una única solución matemática para los flujos internos. Se conocen todos los flujos de la célula a partir de las mediciones del cultivo sin necesidad de optimizar.
GL<0	Sobredeterminado	Redundancia de datos. Hay más mediciones de las estrictamente necesarias. Es el escenario ideal en laboratorios reales, ya que permite aplicar el método de mínimos cuadrados para filtrar errores experimentales y verificar la consistencia del balance de masa.

Capítulo II

Justificación

Problemática

Objetivos

2.1 Justificación

El estudio de *O. alaskensis* resulta de gran relevancia debido a su capacidad de reducir sulfato y participar en procesos de biorremediación de ambientes contaminados por metales pesados, un problema que afecta tanto a la salud humana como a los ecosistemas. Sin embargo, comprender y optimizar su crecimiento requiere enfoques más integrales que combinen microbiología, modelado matemático y herramientas computacionales.

El análisis de rutas metabólicas y el desarrollo de modelos cinéticos permite describir con mayor precisión las mecánicas de crecimiento y desarrollo de *O. alaskensis*. Además, la integración de las bases de datos bioinformáticas fortalece la construcción de modelos confiables y actualizados, permitiendo una contribución al conocimiento fundamental de microorganismos sulfato reductores, el cual es de relevancia en microbiología ambiental.

El uso de herramientas computacionales como Python, Anaconda, Jupyter notebook y paqueterías como pyFOOMB y scikit-learn, proporciona una metodología accesible y reproducible para modelar bioprocesos que permitan implementar análisis de sensibilidad y simulaciones que faciliten la predicción de condiciones óptimas y la identificación de parámetros de mayor influencia para el crecimiento microbiano complementando análisis experimentales (Hemmerich, et al. 2021), esto posiciona la investigación dentro de la biotecnología moderna, donde la computación es un aliado clave del laboratorio.

En términos de biorremediación, *O. alaskensis* ha demostrado su capacidad de precipitar metales pesados en forma de sulfuros insolubles que disminuyen su toxicidad y movilidad en agua y suelos. Por ello, optimizar su crecimiento y metabolismo a través de modelos integrados aporta soluciones viables a problemas ambientales actuales.

2.2 Problemática

Oleidesulfovibrio alaskensis posee un importante potencial biotecnológico en procesos de biorremediación por su capacidad de reducir sulfato y precipitar metales pesados en forma de sulfuros insolubles. Sin embargo, existen limitaciones para comprender cuantitativamente su comportamiento metabólico en relación con la dinámica de crecimiento, el consumo de sustratos y la formación de productos.

Una de las dificultades principales radica en que los estudios disponibles de *O. alaskensis* se enfocan en observaciones experimentales, análisis microbiológicos o descripciones generales de sus capacidades metabólicas, existiendo una limitada integración entre la información bioquímica, datos cinéticos y el modelado matemático estructurado que permita representar de forma mecanística los procesos intracelulares involucrados en la oxidación de lactato y la reducción disimilatoria del sulfato.

Adicionalmente, los modelos cinéticos no estructurados comúnmente empleados en bioprocesos, aunque útiles para predecir tendencias globales de crecimiento y consumo de sustratos, presentan limitaciones para explicar la participación específica de metabolitos intermediarios, enzimas y mecanismos de reacción que permitan optimizar el crecimiento metabólico del sistema.

Por otro lado, la ausencia de metodologías computacionales reproducibles e integradas que combinen herramientas de modelado matemático, análisis de sensibilidad, simulaciones numéricas presentan una barrera para fortalecer la confiabilidad de los resultados obtenidos en estudios de bioprocesos asociados a bacterias sulfato-reductoras.

Ante este contexto, surge la necesidad de desarrollar un modelo cinético estructurado sustentado en rutas metabólicas reconstruidas a partir de bases de datos bioinformáticas y literatura especializada, que permita integrar información experimental, representar de manera más realista la dinámica metabólica de *Oleidesulfovibrio alaskensis* y generar un marco predictivo robusto para comprender su comportamiento y su potencial aplicación en los procesos biotecnológicos.

2.3 Objetivos

Objetivo General

Modelar, simular y validar el crecimiento de *Oleidesulfovibrio alaskensis* mediante el desarrollo e integración de un modelo cinético estructurado para la propuesta de una ruta metabólica integrando herramientas FLOSS con el fin de generar un marco predictivo que fortalezca la comprensión biotecnológica y la aplicación potencial de este microorganismo en procesos de biorremediación.

Objetivos específicos

- Describir e implementar modelos cinéticos no estructurados que representen el crecimiento de *Oleidesulfovibrio alaskensis* bajo condiciones controladas de cultivo, con el fin de establecer parámetros experimentales iniciales que permitan la comparación y validación de modelos estructurados.
- Diseñar e integrar información de rutas metabólicas de *Oleidesulfovibrio alaskensis* a partir de bases de datos bioinformáticas y literatura especializada, para fortalecer la representación del microorganismo y generar un mapa metabólico confiable que sirva como base para la modelación estructurada.
- Construir, describir e implementar un modelo cinético estructurado fundamentado en la ruta metabólica propuesta, que permita simular y analizar la dinámica de crecimiento y eficiencia metabólica del microorganismo, integrando herramientas FLOSS para su validación numérica y experimental.
- Analizar y validar experimentalmente los resultados del modelo no estructurados y estructurado, con el propósito de identificar ventajas, limitaciones y el grado de predictibilidad de cada enfoque en la caracterización del crecimiento microbiano de *Oleidesulfovibrio alaskensis*.

Capítulo III

Metodología

Limitaciones del estudio

3.1 Metodología

La presente investigación desarrolla un modelo dinámico estructurado que integra la ruta metabólica disimilar de reducción de sulfato y la oxidación de lactato en *O. alaskensis*, evaluando su estabilidad, sensibilidad paramétrica e identificabilidad bajo condiciones de cultivo reportadas.

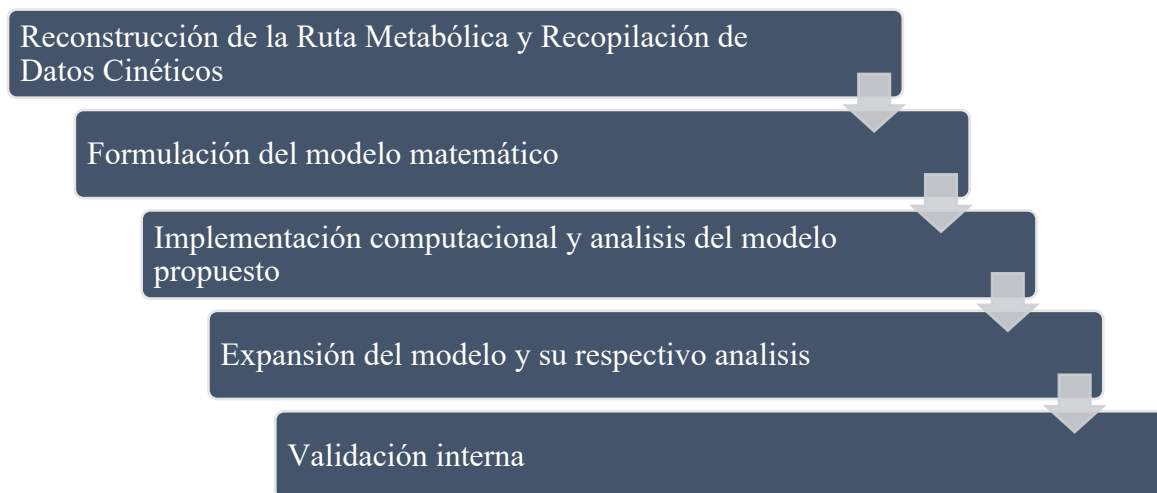


Figura 4. Metodología general.

De acuerdo a la figura 4, la metodología general consiste en la reconstrucción de la ruta metabólica a partir de fuentes bibliográficas y bases de datos para la posterior construcción de un mapa metabólico que servirá como base del desarrollo del modelo matemático propuesto utilizando balances de materia a nivel intracelular y extracelular mediante ecuaciones diferenciales ordinarias. Además, la implementación computacional permitirá analizar la sensibilidad del modelo, para finalmente, expandir el modelo hacia un modelo estructurado que permita un análisis de sensibilidad y selección de parámetros relevantes que permitan la validación intracelular de la ruta metabólica propuesta.

3.1.1 Reconstrucción de la ruta metabólica y recopilación de Datos Cinéticos

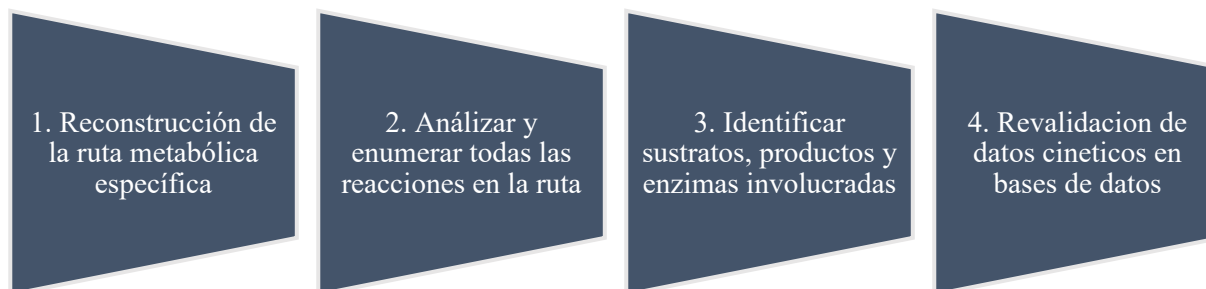


Figura 5. Etapas para la reconstrucción de la ruta metabólica y recopilación de datos cinéticos.

La reconstrucción de la ruta metabólica de *Oleidesulfovibrio alaskensis* se inició con una revisión bibliográfica y en bases de datos metabólicos enfocada en los procesos de oxidación del lactato y la reducción del sulfato. La búsqueda comenzó en el repositorio *KEGG Pathway* con el propósito de identificar las enzimas participantes, los metabolitos intermediarios y las reacciones bioquímicas involucradas. La información preliminar fue complementada con el análisis individual de cada enzima identificada en las bases de datos especializadas, tales como *ExPASy*, *BRENDA*, *MetaCyc*, *NCBI* y *UniProt* con la intención de obtener información sobre parámetros cinéticos reportados, sustratos y productos específicos, cofactores requeridos, condiciones óptimas de funcionamiento o evidencia experimental asociada a cada reacción (Petrovsky et al., 2023).

3.1.2 Formulación del modelo matemático

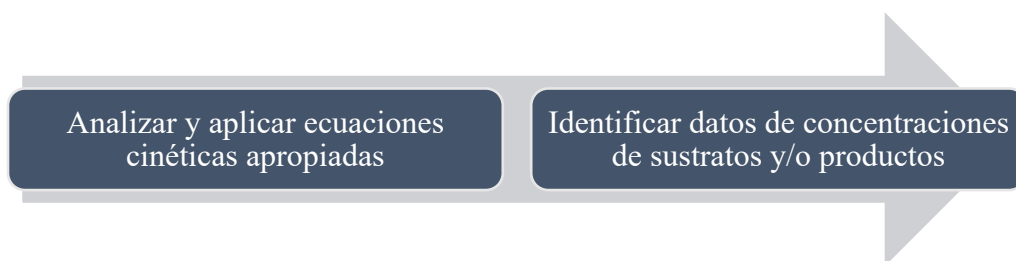


Figura 6. Etapas para la formulación del modelo matemático.

Fuente: López-Pérez (2013).

Con la información recopilada, se construyó un mapa metabólico integrado que describe de forma explícita las rutas de oxidación del lactato a acetato, así como la secuencia de reacciones desde sulfato a sulfuro de hidrógeno. El mapa metabólico constituye la base conceptual para la formulación de un modelo matemático mediante ecuaciones diferenciales ordinarias (EDO), las cuales describen la variación temporal de variables de estado como biomasa, sustratos, productos, y metabolitos intermediarios. La estructura inicial de las ecuaciones se basó en modelos cinéticos previamente desarrollados en trabajos similares de López-Pérez (2013), adaptando términos cinéticos para representar las interacciones metabólicas identificadas en la reconstrucción de las rutas.

3.1.3 Implementación computacional y análisis del modelo propuesto



Figura 7. *Etapas de la implementación computacional y análisis del modelo propuesto.*

Fuente: López-Pérez (2013) y Hemmerich, et al. (2021)

La implementación computacional del modelo se realizó con el lenguaje de programación de *Python*, utilizando el entorno de *Jupyter Notebook* y la paquetería de *pyFOOMB* como marco de modelado de bioprocesos orientado a objetos, esto permite resolver numéricamente los sistemas de ecuaciones planteados, integrar los parámetros procedentes de la literatura y evaluar la estabilidad del modelo bajo distintos escenarios de simulación. Durante esta etapa, los parámetros iniciales se ajustaron con base en los valores cinéticos reportados en trabajos previos, con el objetivo de generar simulaciones comparables en su comportamiento general a modelos de referencia de López-Pérez (2013).

Una vez obtenido un conjunto de simulaciones operativas, se realizó un análisis de sensibilidad paramétrica con la finalidad de evaluar la influencia relativa de cada parámetro sobre las variables de estado del modelo, esto permitió identificar parámetros dominantes y parámetros con baja influencia.

3.1.4 Expansión del modelo y su respectivo análisis

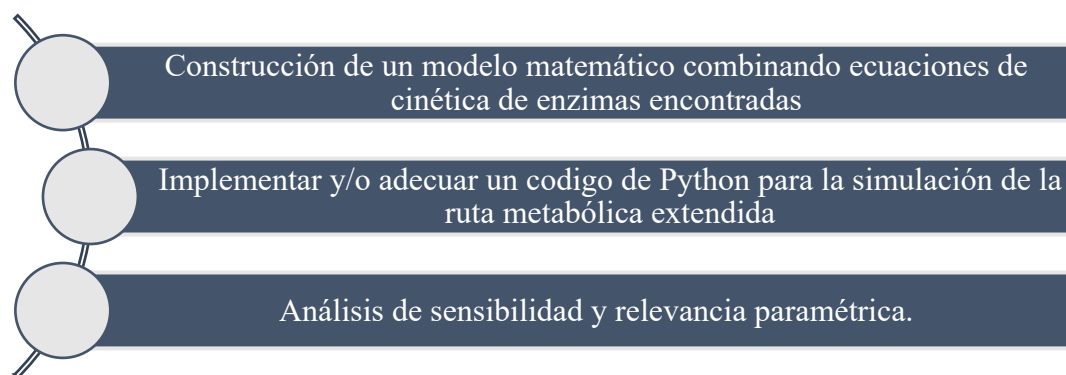


Figura 8. *Pasos a seguir para la expansión del modelo y su respectivo análisis.*

El modelo fue ampliado incorporando ecuaciones asociadas a reacciones enzimáticas para la formación de sulfuro de hidrógeno y la síntesis de acetato de Keller (2014) para integrar un mayor

nivel de detalle metabólico. El objetivo fue evaluar si la inclusión de cinética enzimática mantenía la coherencia de las simulaciones dentro de los rangos de consumo de lactato y sulfato, así como también la producción de acetato, biomasa, CO_2 y H_2S reportados en la literatura de López-Pérez (2013).

Debido al incremento en la dimensionalidad del modelo, se empleó un enfoque complementario de selección de parámetros basados en las herramientas de aprendizaje automático provistas por *Scikit-learn*, que facilitaron la estimación de la importancia relativa de los parámetros en función de su impacto dinámico.

3.1.5 Validación interna

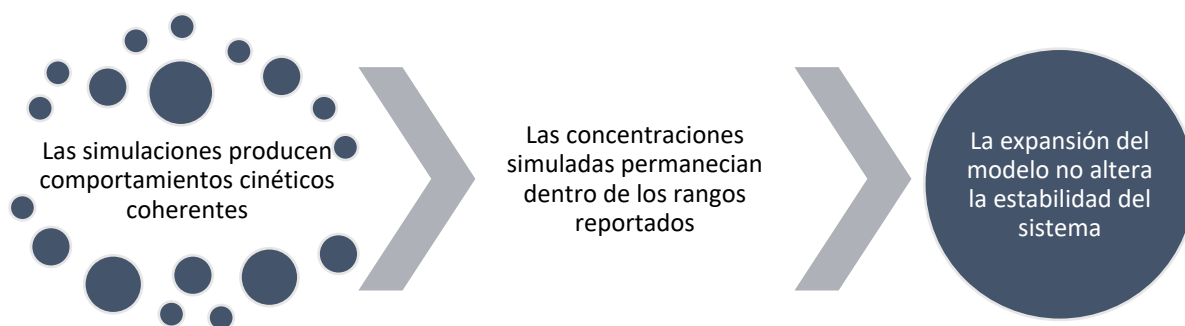


Figura 9. Aspectos importantes de la validación interna del modelo.

La validación del modelo se abordó mediante criterios de coherencia metabólica en el consumo de sustratos y la formación de productos de datos reportados principalmente por Neria-González et al (2010), López-Pérez (2013) y (Jiménez-Perea et al., 2023), estabilidad numérica y concordancia con datos cinéticos (μ_{max} , Y , Y_P , a , b , ks y c) reportados en la literatura anteriormente mencionada.

En primer lugar, se verificó que las simulaciones reprodujeran patrones dinámicos coherentes con el comportamiento esperado de bacterias sulfato-reductoras en aspectos de crecimiento celular, consumo de lactato, formación de acetato, consumo de sulfato, formación de sulfuro y formación de CO_2 , corroborando las concentraciones simuladas de los sustratos y productos permanecieran dentro de los rangos documentados por López-Pérez (2013), (Jiménez-Perea et al., 2023), y los datos del apéndice D.

Así mismo, se observó la robustez del modelo mediante variaciones controladas de parámetros clave, analizando la estabilidad de las soluciones numéricas. La expansión del modelo

con reacciones enzimáticas adicionales fue sometida al mismo procedimiento verificando que el incremento en el detalle metabólico no generara inconsistencias en la dinámica del bioproceso simulado (Coria, s.f.).

El análisis de sensibilidad funcionó como una herramienta complementaria de validación, ya que permitió identificar los parámetros de mayor influencia en el comportamiento del sistema y confirma que la respuesta del modelo es consistente con los principios bioquímicos conocidos (como lo es la dependencia de la velocidad de reacción respecto al sustrato).

3.2 Limitaciones del estudio

El modelo desarrollado representa un avance en la integración de rutas metabólicas y cinética enzimática en bacterias sulfato-reductoras, sin embargo, es importante reconocer ciertas limitaciones relacionadas al enfoque utilizado.

En primera, una parte de los parámetros cinéticos empleados provienen de datos reportados en la literatura para otros organismos cercanos o condiciones experimentales distintas a las específicas de *Oleidesulfobivrio alaskensis*. Si bien esta aproximación es muy común en el modelado metabólico cuando no se dispone de datos experimentales directos, puede incluir incertidumbre en la estimación cuantitativa de algunas tasas de reacción. En el modelo se asumen condiciones homogéneas en el sistema y condiciones de cultivo ideales, sin considerar posibles gradientes espaciales, limitaciones de transporte, ni efectos de la diversidad celular que puede presentarse en sistemas reales, tampoco se incorporaron explícitamente mecanismos regulatorios complejos, como el control transcripcional, la regulación alostérica detallada o respuestas adaptativas al estrés, los cuales son factores que pueden llegar a modificar la actividad enzimática en condiciones cambiantes. Por último, el incremento de detalle estructural del modelo conduce a una mayor complejidad paramétrica, lo cual puede afectar la relevancia de algunos parámetros.

Capítulo IV

Resultados y discusiones

4.1 Modelos cinéticos no estructurados experimentales

Oleidesulfovibrio alaskensis creció en medio Postgate C enriquecido con sulfato de sodio y lactato a un pH de 7 en un reactor por lote, el medio fue ajustado con NaOH (0.5 M), a una temperatura de 37 °C a condiciones anaerobias por 216 horas y se tomaron muestras cada 24 horas para analizar sulfato, sulfuro, biomasa, lactato y dióxido de carbono (López-Pérez, 2013).

Los datos experimentales de la literatura se muestran a continuación en las figuras 10-12 realizadas en Rstudio con la librería de ggplot2 con su respectiva distribución de datos.

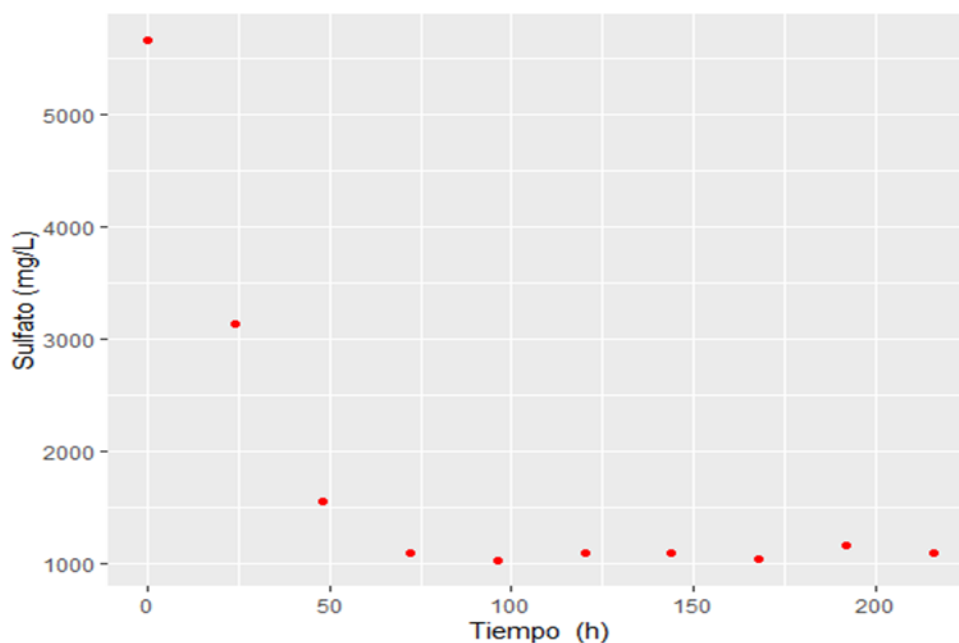


Figura 10. Datos experimentales de sulfato, Rstudio

La desviación estándar es de 1498.676 mg/L, el valor mínimo es de 1034 mg/L, el primer cuartil es de 1103 mg/L, la mediana es de 1103 mg/L, el promedio aritmético es de 1801 mg/L, el tercer cuartil es de 1462 mg/L y el máximo es de 5655 mg/L, todo esto en términos de la concentración de sulfato para 216 h de tiempo de cultivo (Figura 10).

Nota: Adaptado de "Diseño de un bioproceso de alta eficiencia para la remoción de Cadmio vía *Desulfovibrio alaskensis* 6SR" (Tesis doctoral), por P. A. López-Pérez, (2013), Centro de Investigación y de Estudios Avanzados del IPN.

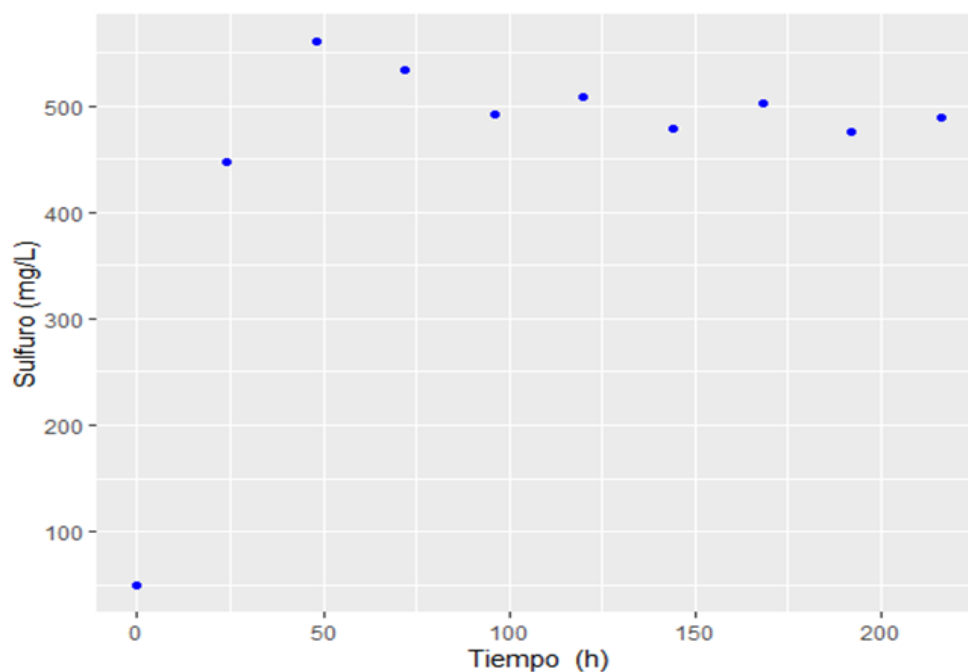


Figura 11. Datos experimentales de sulfuro, Rstudio

La desviación estándar es de 145.1775 mg/L, el valor mínimo es de 50 mg/L, el primer cuartil es de 476.5 mg/L, la mediana es de 490.5 mg/L, el promedio aritmético es de 453.5 mg/L, el tercer cuartil es de 506.5 mg/L y el máximo es de 560 mg/L, todo esto en términos de la concentración de sulfuro para 216 h de tiempo de cultivo (Figura 11).

Nota: Adaptado de "Diseño de un bioproceso de alta eficiencia para la remoción de Cadmio vía *Desulfovibrio alaskensis* 6SR" (Tesis doctoral), por P. A. López-Pérez, (2013), Centro de Investigación y de Estudios Avanzados del IPN.

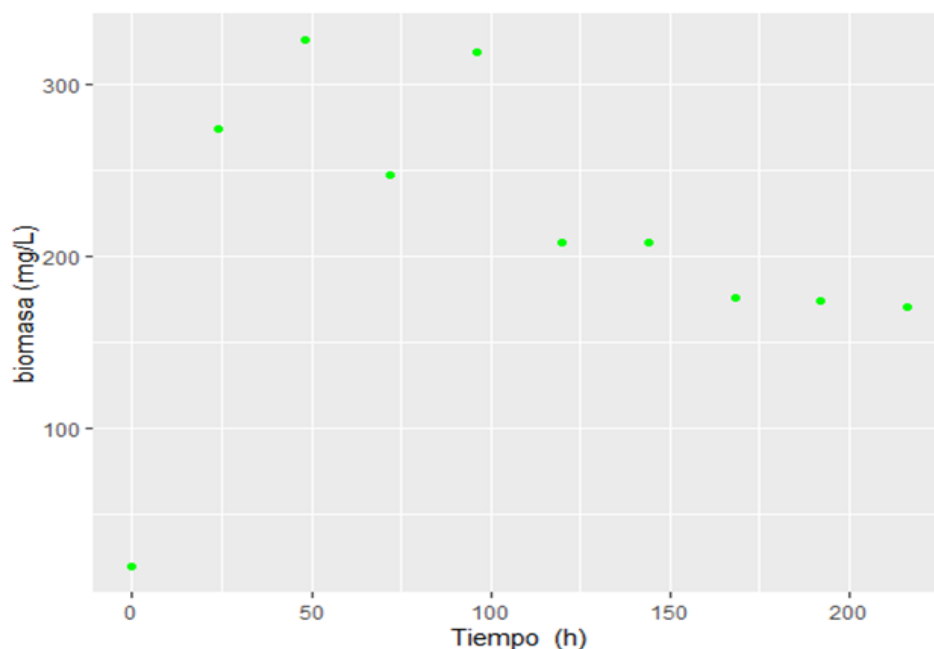


Figura 12. Datos experimentales de biomasa, Rstudio.

La desviación estándar es de 88.7571 mg/L, el valor mínimo es de 20 mg/L, el primer cuartil es de 174.5 mg/L, la mediana es de 208 mg/L, el promedio aritmético es de 212.4 mg/L, el tercer cuartil es de 267.5 mg/L y el máximo es de 326 mg/L, todo esto en términos de la concentración de sulfuro para 216 h de tiempo de cultivo (Figura 12).

Por último, la desviación estándar es de 72.66361 h, el valor mínimo es de 0 h, el primer cuartil es de 54 h, la mediana es de 108 h, el promedio aritmético es de 108 h, el tercer cuartil es de 162 h y el máximo es de 216 h, todo esto en términos del tiempo de cultivo.

Nota: Adaptado de "Diseño de un bioproceso de alta eficiencia para la remoción de Cadmio vía *Desulfovibrio alaskensis* 6SR" (Tesis doctoral), por P. A. López-Pérez, (2013), Centro de Investigación y de Estudios Avanzados del IPN.

Tabla 6. *Parámetros cinéticos reportados en la literatura para el diseño del modelado cinético.*

Parámetros cinéticos de crecimiento reportados en literatura				
Parámetro	Definición	Valor	Unidades	Fuente
S1	Es la concentración máxima de producto (saturación o inhibición)	680	mgL ⁻¹	López-Pérez (2013)
k_{ace}	Es la constante de saturación para acetato	1500	mgL ⁻¹	López-Pérez (2013)
k_{lac}	Es la constante de saturación para lactato	95000	mgL ⁻¹	López-Pérez (2013)
k_s	Es la constante de saturación para el sustrato	0.0009048	mgL ⁻¹	Neria-González et al (2010) López-Pérez (2013)
k_{co}	Es la constante de saturación para el dióxido de carbono	900	mgL ⁻¹	López-Pérez (2013)
k_d	Es la constante de tasa de muerte celular	0.062 0.0027	h ⁻¹	Neria-González et al (2010) (Jiménez-Perea et al., 2023)
μ_{max}	Es la velocidad máxima de crecimiento microbiano	0.188 0.1014	h ⁻¹	Neria-González et al (2010) López-Pérez (2013) (Jiménez-Perea et al., 2023)
μ_{max3}	Es la velocidad máxima de consumo de lactato	1.8	h ⁻¹	López-Pérez (2013)
μ_{max5}	Es la velocidad máxima de producción de dióxido de carbono	0.12	h ⁻¹	López-Pérez (2013)
Y	Es la tasa de rendimiento de biomasa a partir del sustrato	14.13 1 7.4443	mgS mg ⁻¹ X g/g	Neria-González et al (2010) López-Pérez (2013) (Jiménez-Perea et al., 2023)
YP	Es la tasa de rendimiento del producto a partir del producto	2.14 7.5 1.7068	mgP mg ⁻¹ X g/g	Neria-González et al (2010) López-Pérez (2013) (Jiménez-Perea et al., 2023)

a	Es el coeficiente de inhibición por producto	0.89 1.9	Adimensional	Neria-González et al (2010) López-Pérez (2013)
b	Es el efecto del lactato en el consumo de sustrato	3.1	Adimensional	López-Pérez (2013)
c	Es la no linealidad en el consumo de lactato	0.001	Adimensional	López-Pérez (2013)
μ_{max1}	Es la velocidad máxima de formación de producto	0.156	mM s ⁻¹	Lamperia et al (1990)
k_b	Es la constante de transformación de sustrato a producto	0.6 3.2 0.82	s ⁻¹	Lamperia et al (1990)
μ_{max2}	Es la velocidad máxima de formación de producto	0.0055	mM s ⁻¹	En este trabajo
k_{d2}	Es la constante de transformación de sustrato a producto	0.31	s ⁻¹	Wolfe et al (1994)
μ_{max4}	Es la velocidad máxima de formación de producto	14.8	$\mu\text{mol}/\text{min}/\text{mg}$	Furdi et al (2000)
k_e	Es la constante de transformación de sustrato a producto	2.0	s ⁻¹	Furdi et al (2000)
μ_{max6}	Es la velocidad máxima de formación de producto	177.4 23.1	U/mg	Campos et al (2010)
k_f	Es la constante de transformación de sustrato a producto	227.6 29.6	s ⁻¹	Campos et al (2010)
μ_{max7}	Es la velocidad máxima de formación de producto	68.0	U/mg	Janson et al (1970)
k_g	Es la constante de transformación de sustrato a producto	34	$\mu\text{M s}^{-1}$	Janson et al (1970)

Nota: Para más información de las enzimas utilizadas consulte el Apéndice C. *Fuente:* elaboración propia.

La tabla 6 es el resultado de la búsqueda de valores para los parámetros cinéticos utilizados en el modelado no estructurado y en el modelado estructurado. Estos valores se sustituyeron en los posteriores balances de materia (ecuaciones 1 a 17).

4.1.1 Modelo matemático no estructurado para operación por lote

Balance para la concentración de sulfato (S):

$$\frac{dS}{dt} = -Y \cdot \mu_{max} \left[1 - \left(\frac{P}{S_1} \right) \right]^a \left[\frac{S}{k_{S+S}} \right] X \cdot L^b \quad (1)$$

Balance para la concentración de sulfuro (P):

$$\frac{dP}{dt} = (\mu_{max} \cdot YP) \left[1 - \left(\frac{P}{S_1} \right) \right]^a \left[\frac{S}{k_{S+S}} \right] X \cdot L^b \quad (2)$$

Balance para la concentración de lactato (L):

$$\frac{dL}{dt} = -(\mu_{max_3}) \left[\frac{k_{ace}}{A+k_{ace}} \right] \left[\left(\frac{L^c}{k_{lac+L^c}} \right) \right] X \quad (3)$$

Balance para la concentración de acetato (A):

$$\frac{dA}{dt} = (\mu_{max_3}) \left[\frac{k_{ace}}{A+k_{ace}} \right] \left[\left(\frac{L^c}{k_{lac+L^c}} \right) \right] X \quad (4)$$

Balance para la concentración de dióxido de carbono (C):

$$\frac{dC}{dt} = (\mu_{max_5}) \left[\frac{k_{co}}{C+k_{co}} \right] \left[\left(\frac{L^c}{k_{lac+L^c}} \right) \right] X \quad (5)$$

Balance para la concentración de biomasa (X):

$$\frac{dX}{dt} = (-\mu_{max}) \left[1 - \left(\frac{P}{S_1} \right) \right]^a \left[\left(\frac{L^c}{k_{lac+L^c}} \right) \right] X - k_d \cdot X \cdot L^b \quad (6)$$

Las ecuaciones de la (1) a la (6) describen la tasa de crecimiento específico bacteriano (μ) y la concentración de sustrato $[S]$. El modelo anterior considera el efecto de inhibición por Cd^{2+} y H_2S en el crecimiento bacteriano. Incluye dos aspectos importantes, la reducción del sulfato por el cultivo microbiano y el consumo de la fuente de carbono (lactato para este caso). De igual manera, se describe un crecimiento clásico de reducción de sulfato a sulfuro. Además, un modelo de inhibición de Levespiel fue modificado para describir los efectos de la reducción del sulfato con la inhibición por H_2S durante el crecimiento bacteriano en la ecuación (2). Por último, se estimó la producción de CO_2 y la formación de acetato $[A]$ a partir del consumo de Lactato $[L]$ con los modelos combinados de Moser-Boulton y la concentración de biomasa $[X]$ en las ecuaciones (1, 3, 4, 6, 5), tal como lo describe López-Pérez (2013).

Tabla 7. *Variables del modelo no estructurado.*

Nombre de la variable	Nomenclatura en el modelo
Sulfato	S
Sulfuro de hidrógeno	P
Biomasa	X
Lactato	L
Acetato	A
CO₂	C

El modelo considera concentraciones iniciales de $[S_o] = 5655$ mg/L; $[P_o] = 29$ mg/L; $[X_o] = 20$ mg/L; $[L_o] = 4640$ mg/L; $[A_o] = 0.1$ mg/L; $[C_o] = 0.1$ mg/L. Así mismo, considera parámetros cinéticos de $S_I = 680$; $k_{ace} = 1500$; $k_{lac} = 95000$; $k_s = 0.0009048$; $k_{co} = 900$; $k_d = 0.04$; $\mu_{max} = 0.13$; $\mu_{max_3} = 0.6$; $\mu_{max_5} = 0.12$; $Y = 7.5$; $YP = 1$; $a = 1.9$; $b = 0.001$; $c = 1.5$.

4.1.1.1 Algoritmos

El código utilizado para modelar utiliza la biblioteca de pyFOOMB (Python Framework for Object-Oriented Modeling of Bioprocesses), el cual permite simular procesos biológicos que muestre la evolución temporal de las concentraciones de lactato, acetato, sulfato, sulfuro de hidrógeno, biomasa y CO₂. El modelo describe una cinética con múltiples términos regulatorios mediante las 6 ecuaciones ordinarias (EDOs) no lineales y acopladas, 13 parámetros en los cuales se consideran constantes cinéticas, velocidades, rendimientos y exponentes; las condiciones iniciales descritas anteriormente (Figura 13).

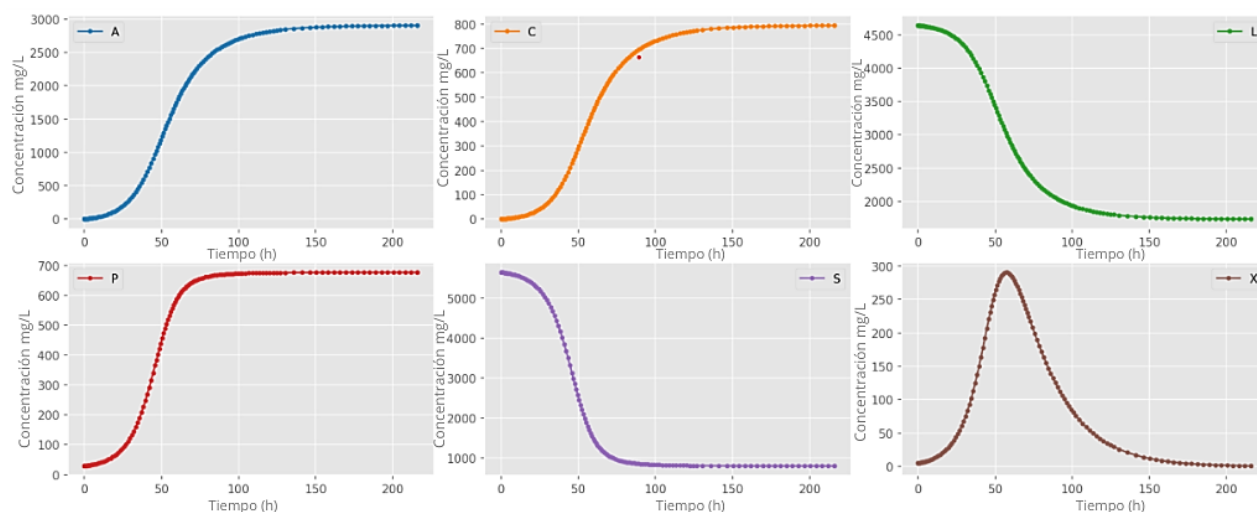


Figura 13. Cinética de caja negra obtenida con el paquete pyFOOMB.

Este modelo representa la fermentación láctica por *O. alaskensis*, donde la biomasa (X) crece a partir del sustrato (S), la producción de producto (P) está acoplada al crecimiento, el lactato (L) se consume para activar la producción y la formación de acetato (A) y Dióxido de carbono (C) son subproductos del metabolismo.

NOTA: La línea azul A corresponde a la formación de acetato con respecto al tiempo, la línea naranja C corresponde a la formación de dióxido de carbono con respecto al tiempo, la línea verde L corresponde al consumo de sustrato con respecto al tiempo, la línea roja P corresponde a la formación de sulfuro con respecto al tiempo, la línea morada S corresponde al consumo de sulfato con respecto al tiempo y la línea café X corresponde a la formación de biomasa con respecto al tiempo. Todas las gráficas asemejan el comportamiento real de una cinética de *Oleidesulfobivrio alaskensis*.

Al momento de importar la librería pyFOOMB, se importan tres clases fundamentales: 1) **BioprocessModel**: que es la clase base para definir modelos de bioprocesos; 2) **Caretaker**: que es el gestor que maneja el modelo, sus parámetros y simulaciones; y 3) **Visualization**: es de utilidad para graficar los resultados. Posteriormente, se define una clase **SequentialKinetic** que se hereda de **BioprocessModel**. El método clave es **rhs** (apéndice A, código 1), que calcula las derivadas donde t es el tiempo, y $\mathbf{y}$ es el vector de estados (para A, L, S, P, X y C), los parámetros se obtienen del diccionario **model_parameters** del modelo y se retorna una lista con las seis derivadas. Además, se tienen que definir los valores iniciales y los parámetros en diccionarios. El **Caretaker** se encarga de administrar el modelo y sus parámetros permitiendo crear una instancia del modelo con la parametrización dada, simular el sistema para un tiempo determinado y manejar múltiples réplicas (instancias) del mismo modelo con diferentes parámetros. Luego se pueden realizar diferentes simulaciones con el método **simulate** para que devuelva un objeto con los resultados (tiempos y trayectorias de los resultados). La clase **Visualization** proporciona un método **show_kinetic_data** para graficar las curvas de concentración vs. tiempo, lo que permite generar figuras para cada concentración, mostrando su evolución a través del tiempo.

La figura 13 muestra la evolución temporal de las concentraciones de los principales compuestos involucrados en el bioproceso; acetato (A), dióxido de carbono (C), lactato (L), sulfuro (P), sulfato (S) y biomasa (X) durante una operación por lote. Se observa que el lactato (L) es consumido y utilizado como fuente de carbono y energía al tener una disminución progresiva a lo largo del tiempo. De manera simultánea, el sulfato (S) disminuye considerablemente al actuar como aceptor final de electrones, lo que confirma su reducción en el proceso disimilatorio. Además, se evidencia la formación de productos metabólicos. El acetato (A) y el dióxido de carbono (C) muestran un crecimiento sigmoideal hasta alcanzar una fase estacionaria. De forma similar, el sulfuro (P) aumenta con el tiempo como producto de la reducción del sulfato. Por otro lado, la biomasa (X) presenta un crecimiento inicial hasta alcanzar un punto máximo seguido de una fase de declive. Este comportamiento sugiere una fase exponencial limitada por la disponibilidad de sustratos, seguida por una fase de decaimiento posiblemente asociada al agotamiento de lactato y/o sulfato, así como la acumulación del producto inhibitorio (el sulfuro).

4.1.1.2 Sensibilidad

Para este análisis se empleó una ampliación en el código anterior para introducir una función muy importante: sensibilidad de los parámetros. El propósito consiste en analizar cómo los parámetros influyen en las trayectorias de los estados, esto se hace mediante sensibilidades entre las derivadas parciales de cada variable de estado con respecto a cada parámetro. Las sensibilidades indican que parámetros tienen mayor impacto en la dinámica del sistema, esta información suele ser clave para:

- 1) Calibración del modelo: identificando qué parámetros pueden estimarse con mayor precisión a partir de datos experimentales.
- 2) Diseño de experimentos: para saber qué variables medir para obtener información sobre parámetros específicos.
- 3) Análisis de identificabilidad: para detectar parámetros correlacionados o que apenas afectan al modelo.

Este nuevo código ampliado, además de las clases ya conocidas (`BioprocessModel`, `Caretaker` y `Visualization`), se importan dos nuevas:

- 1) `ObservationFuncion`: que permite definir funciones de observación (relacionando estados con mediciones).
- 2) `Measurement`: maneja datos experimentales (aunque en este código no se usen directamente).

Después se establece una semilla para el generador de números aleatorios (`numpy.random.seed(123)`) para garantizar la reproductibilidad en caso de que se generen datos sintéticos o se realicen perturbaciones aleatorias.

En resumen, este código permite modelar un sistema de reacciones acopladas con múltiples estados y parámetros, simular su comportamiento dinámico, calcular sensibilidades para entender qué parámetros tienen mayor impacto en las predicciones del modelo, seleccionar parámetros para experimentos de calibración o diseño y comparar diferentes métodos de aproximación (diferencia relativa o absoluta).

Tabla 8. Sensibilidad de los parámetros.

Parámetros	S	P	L	A	C	X
SI	<	>	>	>	>	>
K_{ace}	>	<	>	>	<	>
K_{lac}	>	>	>	>	<	<
K_s	>	>	>	<	>	<
K_{co}	<	<	<	<	>	>
K_d	>	>	>	>	<	>
μ_{max}	>	>	>	>	>	>
μ_{max_4}	<	>	>	>	>	>
μ_{max_5}	<	>	<	>	>	>
Y	>	>	<	>	<	>
YP	>	>	>	<	>	>
A_0	>	<	>	<	<	<
C_0	<	<	<	<	<	<
L_0	<	<	<	>	>	<
P_0	>	>	>	>	<	<
S_0	>	>	<	>	>	>
X_0	>	>	>	>	>	>

Símbolo	Sensibilidad
>	Muy alta
>	Alta
<	Moderada
>	Débil
<	Despreciable

El análisis de sensibilidad arrojó un total de 102 figuras de parámetro y variable (figura A 30, Apéndice A). Los resultados se resumen en la tabla 8 donde se muestra que los parámetros μ_{max} y k_d influyen de manera significativa en todas las variables del sistema, esto es indicativo de que la dinámica global está fuertemente controlada por la tasa de crecimiento y los procesos de inhibición. YP presenta muy alta sensibilidad en las variables S , P , L , C y X , lo que lo vuelve un parámetro clave en la formación de productos. Por otro lado, las concentraciones iniciales de biomasa (X_0) afecta fuertemente todas las variables, esto confirma su impacto temprano en la evolución del bioproceso desde etapas tempranas (Tabla 8).

A partir de los resultados de la tabla 8 se realizó una conversión a valores numéricos (material complementario) para obtener el mapa de calor realizado con Rstudio de la figura 14, donde se observa las relaciones entre parámetros y las relaciones entre variables.

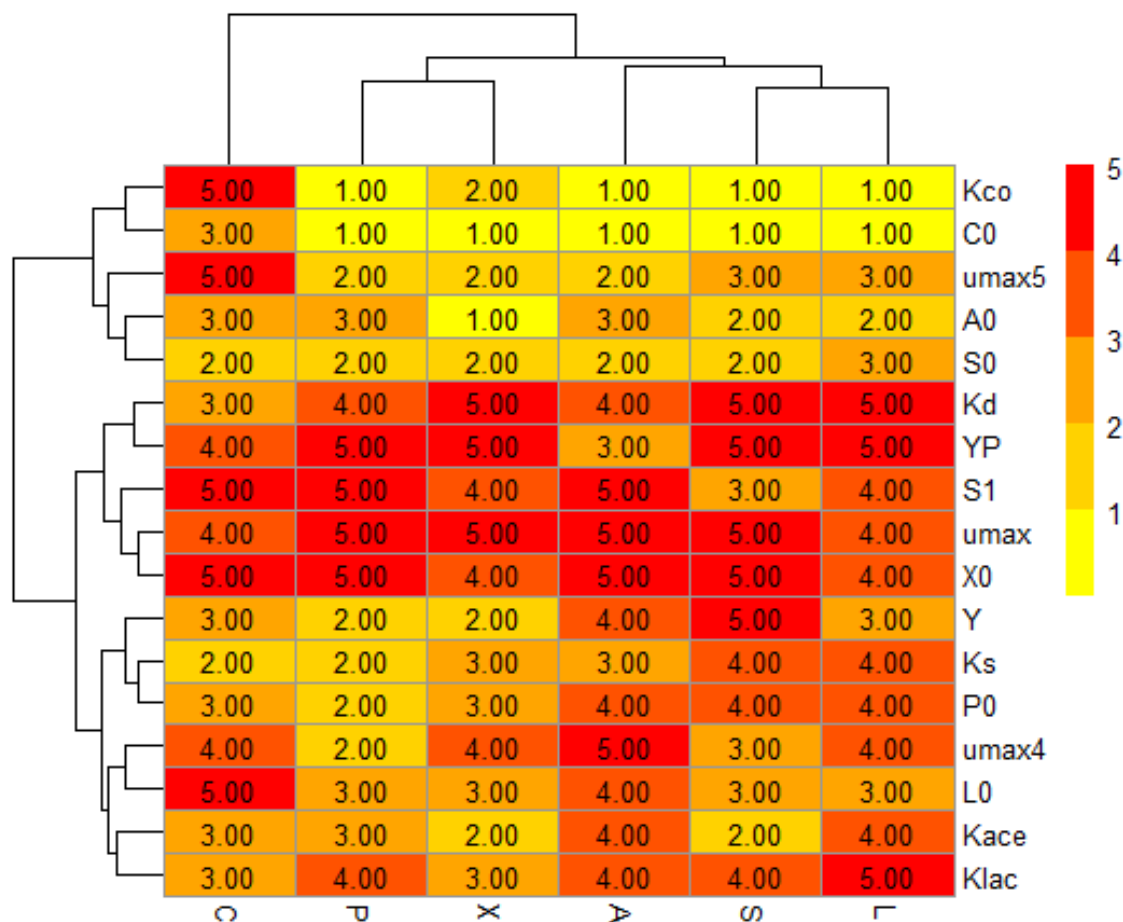


Figura 14. Mapa de calor de sensibilidades paramétricas del modelado de caja negra.

Fuente: elaboración propia

El mapa de calor (Figura 14) muestra que los parámetros μ_{max} , Y , $S1$, YP y $X0$ (con valores altos de 4-5 en múltiples variables) son los que principalmente afectan de manera simultánea el crecimiento de biomasa, producto y consumo de sustrato. Los parámetros Kco , Co y Ks muestran baja sensibilidad, lo que sugiere que tienen un rol secundario en la ruta metabólica. El clustering de variables indica que algunas variables comparten dependencia paramétrica y que existen módulos dinámicos agrupados, es decir, hay acoplamiento entre la reducción de sulfato y el crecimiento de biomasa y dependencia compartida del flujo electrónico.

4.1.1.3 Incertidumbre

Para esta sección, se implementa un nuevo código basado en la ampliación del código en la sección de modelado, anteriormente descrita, con el propósito de realizar una estimación de Monte Carlo. Se pretende cuantificar la incertidumbre en los parámetros del modelo disponiendo de los datos experimentales con ruido para que, en lugar de obtener una única estimación (que daría un valor puntual), se generan múltiples conjuntos de datos sintéticos con ruido y se estiman los parámetros de cada uno y así, obtener una forma de análisis no lineal de incertidumbre.

Se define una clase `SequentialKinetic` de `BioprocessModel` para describir un cultivo microbiano con seis variables de estado: Lactato (L), acetato (A), Sulfato (S), Sulfuro (P), Biomasa (X) y Dióxido de carbono (C). Las ecuaciones que determinan el sistema son de la ecuación 1 a 6. Los parámetros del modelo (constantes) se definen en un diccionario (`model_parameters`) y los valores iniciales en `inicial_values`, para crear el objeto `Caretaker` que gestionará el modelo y las simulaciones. El método `caretaker.estimate` realiza la optimización de parámetros. Por defecto una un optimizador global sin derivadas (differential evolution). Se muestran los resultados y se guardan en `est` (parámetros estimados) y `est_info` (información del optimizador), luego se visualiza la comparación entre los datos experimentales y la simulación con los datos estimados, junto con la simulación con los parámetros verdaderos.

Posteriormente, la función `noisy_samples` toma una simulación (objeto `TimeSeries`) y genera mediciones artificiales, selecciona aleatoriamente unos pocos puntos de tiempo, añade ruido gaussiano con error relativo y absoluto específicos y repite el proceso varias veces y lo promedia para simular réplicas. Con ayuda del `Caretaker` se realiza una simulación del modelo con los parámetros verdaderos y se aplica `noisy_samples` para obtener datos con ruido como se muestra a continuación en la figura 15.

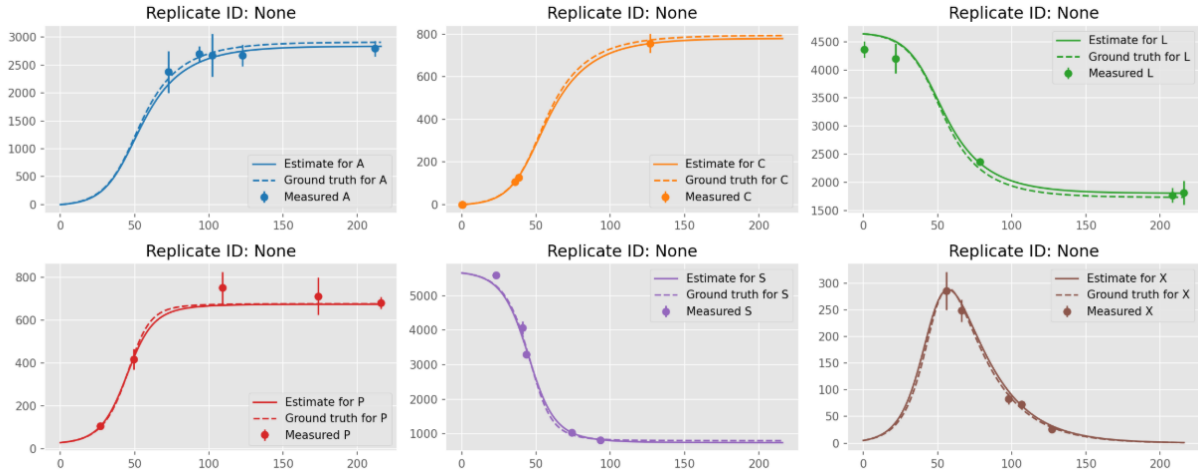


Figura 15. Cinética no estructurada de *Oleidesulfovibrio alaskensis* con distribuciones de los parámetros estimados con pyFOOMB.

El método clave es `caretaker.estimate_parallel_MC_sampling`, ya que realiza lo siguiente:

- 1) Muestreo de Monte Carlo: aquí se generan múltiples realizaciones del conjunto de datos de entrenamiento, cada una con diferente ruido (usando la función `noisy_samples` internamente), para cada realización se ejecuta una estimación de parámetros completa.
- 2) Configuración: aquí se especifican los parámetros a estimar (`unknowns`) y sus límites (`bounds`). También se puede ajustar el número de islas (poblaciones) y los optimizadores utilizados.
- 3) Convergencia: cada estimación individual finaliza cuando se alcanza una tolerancia relativa (`rtol_islands`) o tras un número máximo de evoluciones.

Dando como resultado final un gráfico tipo *corner plot* (apéndice B) que muestra las distribuciones marginales y correlaciones entre parámetros, junto con coeficientes de correlación final.

Nota: Comparación entre datos simulados y experimentales para las concentraciones de acetato (*A*), dióxido de carbono (*C*), lactato (*L*), sulfuro (*P*), sulfato (*S*) y biomasa (*X*) en función del tiempo. Las líneas continuas representan los valores estimados por el modelo, las líneas punteadas corresponden a los valores teóricos (ground truth), y los puntos indican los datos experimentales con sus respectivas barras de error. En general, se observa un buen ajuste del modelo para todas las variables, reproduciendo adecuadamente las tendencias de consumo de sustratos (*L* y *S*) y la formación de productos (*A*, *C* y *P*), así como la dinámica de decaimiento de la biomasa (*X*).

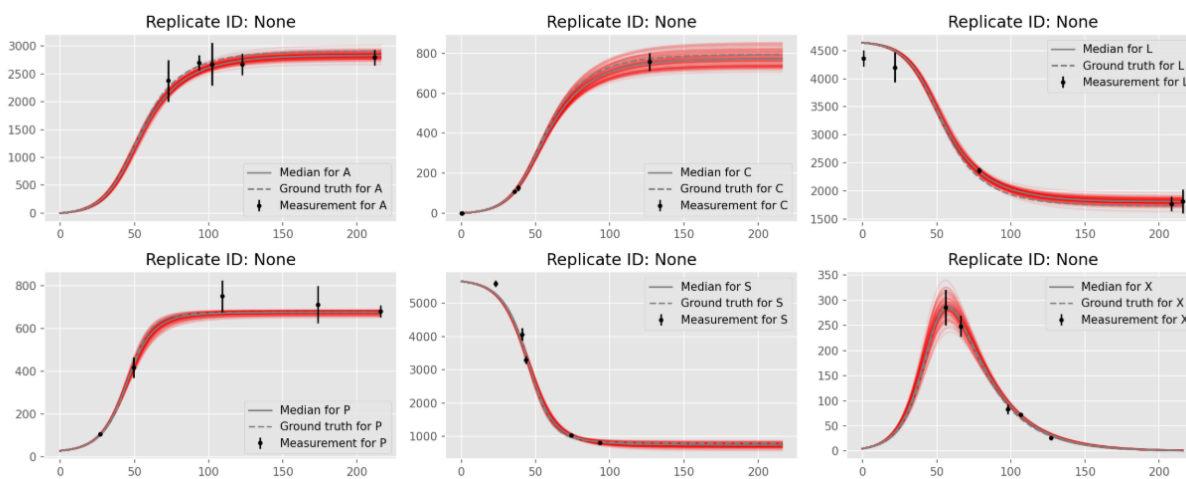


Figura 16. Cinética no estructurada de *Oleidesulfovibrio alaskensis* donde se comparan las simulaciones de los diferentes conjuntos de parámetros con pyFOOMB.

Una vez obtenidas las múltiples estimaciones, se pueden calcular estadísticos descriptivos (media, desviación estándar, percentiles) para cada parámetro, visualizar las distribuciones empíricas mediante histogramas o diagramas de caja y comparar las estimaciones con los valores verdaderos (es decir, los resultados para generar los datos sintéticos) para evaluar el sesgo y la precisión.

El análisis de correlación entre parámetros estimados mediante muestreo Monte Carlo revela que las correlaciones fuertes identificadas son Y vs YP con un valor de $\rho = 0.971$ (correlación positiva), que implica que los rendimientos de sulfato y sulfuro están estrechamente acoplados; μ_{max} vs Y con un valor de $\rho = -0.801$ (correlación negativa) que sugiere una compensación entre la tasa de crecimiento y eficiencia de conversión; kd vs μ_{max} con un valor de $\rho = 0.892$ (correlación positiva) que sugiere que mayores tasas de crecimiento se asocian con mayor mortalidad, posiblemente por estrés metabólico.

Nota: Comparación entre datos simulados y experimentales para las concentraciones de acetato (A), dióxido de carbono (C), lactato (L), sulfuro (P), sulfato (S) y biomasa (X) en función del tiempo. La línea negra continua representa la mediana de las simulaciones del modelo, mientras que la banda roja indica la variabilidad asociada a la estimación de los parámetros. La línea punteada corresponde a los valores de referencia (ground truth) y los puntos negros representan las mediciones experimentales con sus respectivas barras de error. En general, el modelo captura adecuadamente la dinámica de consumo de sustratos (L y P), la formación de productos (A , C , y P) y el comportamiento de crecimiento y decaimiento de la biomasa (X), mostrando además una incertidumbre acotada en las predicciones.

Los parámetros altamente correlacionados deben calibrarse simultáneamente, las correlaciones positivas sugieren mecanismos regulatorios compartidos y las correlaciones negativas indican trade-offs metabólicos, un principio biológico y de ingeniería que describe la relación inversa donde la maximización de la velocidad del producto se logra a expensas de la eficiencia total en rendimiento o biomasa final como lo describe en sus estudios Frank (2010), Lipson (2015) sugiere que bioenergéticamente, dependen de la disponibilidad de recursos, si el objetivo es lograr un crecimiento rápido, los microorganismos utilizarán las vías metabólicas que gastan más energía (eficiencia energética menor) en lugar de optimizar la conversión de recursos para generar biomasa. Sin embargo, Manhart et al. (2018) sugiere que, dependiendo del microorganismo y el entorno donde se desarrolle, las correlaciones del crecimiento vs rendimiento pueden indicar que el microorganismo se adapta a entornos inestables maximizando la tasa de crecimiento o que maximizan el rendimiento en entornos estables. Dicho esto, y de acuerdo con los resultados, se puede concluir que *O. alaskensis* utiliza su metabolismo para un crecimiento rápido (alta μ_{max}) a costa de una menor eficiencia en la conversión de sustrato (bajo Y).

Cheng et al. (2019) describe que el metabolismo de desbordamiento puede estar acoplado o desacoplado de la tasa de crecimiento, esto se traduce a que el trade-off es de naturaleza bidimensional, donde el balance no depende de uno, sino de dos factores distintos que limitan el crecimiento, lo que es evidente en este modelo al existir la correlación negativa entre el crecimiento vs rendimiento y al identificarse μ_{max} y b (asociado al lactato) como parámetros dominantes. La tabla 9 muestra un resumen de las distribuciones marginales y correlaciones entre parámetros más relevantes, obtenida a partir de los resultados encontrados en el material complementario en el apéndice C, además de incluir una recomendación para futuras calibraciones (otro estudio no desarrollado aquí) sobre el modelo.

Tabla 9. Resumen de los resultados de las distribuciones marginales, correlaciones entre parámetros, y coeficientes de correlación final.

Parámetro	Identificación	Correlación	Recomendación
S1	Alta (distribución concentrada)	Baja	Usar directamente en el modelo.
k_s	Alta	Baja	Confiable, se debe mantener en la calibración.

k_{lac}	Alta	Baja	Bien definido, no necesita fijarse.
μ_{max}	Baja (dispersión amplia)	Fuerte con Y (negativa)	Fijar con literatura o validar con más datos experimentales.
Y	Baja (dispersión amplia)	Fuerte con μ_{max}	Igual que μ_{max} , fijar o acotar con más experimentos.
k_{ace}	Media-baja (histograma ancho)	Moderada con otros parámetros	Puede fijarse en un rango reportado en literatura.
Otros parámetros	Media-baja	Algunas correlaciones entre sí	Mantener sólo si afectan significativamente la salida del modelo; de lo contrario fijarlos.

Además, la tabla 8 permitió identificar que los parámetros SI (concentración máxima de producto), μ_{max} (tasa máxima de crecimiento), kd (constante de muerte), Y y YP (rendimientos) ejercen la mayor influencia sobre las variables del sistema. En particular, SI mostró alta sensibilidad sobre la producción de sulfuro (P), lactato (L), acetato (A) y biomasa (X), lo que sugiere que la inhibición por H_2S (P) es un factor limitante dominante, en concordancia con reportes previos de Keller (2014). La alta sensibilidad de μ_{max} y kd refleja la importancia de la eficiencia energética y la tasa de recambio celular en la dinámica del cultivo.

4.2 Propuesta de ruta metabólica

4.2.1 Recopilación de información.

A partir de la información disponible de KEGG sobre el metabolismo del sulfato de *O. alaskensis* se realizó una búsqueda en bases de datos y artículos científicos para confirmar y complementar la tabla 3 del apartado de antecedentes. El resultado se encuentra en las tablas de la 10 a la 15 que se muestran a continuación.

Tabla 10. Reacción bioquímica de la ruta de lactato a acetato mediante bases de datos.

RUTA DE LACTATO A ACETATO		
Enzimas	Reacción	Fuente
D-lactato deshidrogenasa (citocromo)	(R)-lactato + una quinona $\rightleftharpoons$ un quinol + piruvato	UniPro (1)*
Piruvato-ferredoxina oxidoreductasa	CoA + 2[2Fe-2S]-[ferredoxina] oxidada + piruvato $\rightleftharpoons$ acetil-CoA + CO ₂ + H ⁺ + 2[2Fe-2S]-[ferredoxina] reducida	UniProt (2)*
Fosfotransacetilasa	Acetil-CoA + Fosfato $\rightleftharpoons$ Acetilfosfato + CoA	UniProt (3)*
Acetato quinasa	Acetilfosfato + ADP $\rightleftharpoons$ Acetato + ATP	UniProt (4)*

Nota: link de la base de datos con la información.

(1)* [D-lactate dehydrogenase \(cytochrome\) - *Oleidesulfovibrio alaskensis* \(strain ATCC BAA-1058 / DSM 17464 / G20\) \(Desulfovibrio alaskensis\) | UniProtKB | UniProt](#)

(2)* <https://www.uniprot.org/uniprotkb/Q30WB5/entry>

(3)* <https://www.uniprot.org/uniprotkb/Q30WB1/entry>

(4)* <https://www.uniprot.org/uniprotkb/Q30WB0/entry>

Tabla 11. *Reacción bioquímica de la ruta de sulfato a sulfuro mediante bases de datos.*

ruta de sulfato a sulfuro		
Enzimas	Reacción	Fuente
ATP sulfurilasa	$\text{ATP} + \text{H}^+ + \text{sulfate} \rightleftharpoons \text{adenosina 5'-fosfosulfato} + \text{difosfato}$	UniProt (5)*
Pirofosfatasa	$\text{Difosfato} + \text{H}_2\text{O} \rightleftharpoons \text{H}^+ + 2 \text{ fosfato}$	UniProt (6)*
Adenosina-5'-fosfosulfato (APS) reductasa	$\text{Adenosina-5'-fosfosulfato (APS)} + 2 \text{ Electrones (ferredoxina o glutaredoxina reducida)} \rightleftharpoons \text{Sulfito} + \text{AMP}$	BRENDA (7)*
[DsrC]-trisulfuro reductasa	$\text{sulfuro de hidrógeno} + \text{un [proteína DsrC]-ditiol} + 2 \text{ quinonas} \rightleftharpoons \text{un [proteína DsrC]-trisulfuro} + 2 \text{ quinoles}$	IUBMB (8)*
Bisulfito reductasa	$[\text{Proteína DsrC}]\text{-trisulfuro} + 3 \text{ H}_2\text{O} + \text{NAD}(+) \rightleftharpoons [\text{Proteína DsrC}]\text{-ditiol} + 3 \text{ H}(+) + \text{NADH} + \text{sulfito}$	Expasy (9)*

(5)* <https://www.uniprot.org/uniprotkb/Q30Z34/entry>

(6)* <https://www.uniprot.org/uniprotkb/Q310S1/entry>

(7)* <https://www.brenda-enzymes.info/enzyme.php?ecno=1.8.99.2>

(8)* <https://iubmb.qmul.ac.uk/enzyme/EC1/8/5/10.html>

(9)* <https://enzyme.expasy.org/EC/1.8.1.22>

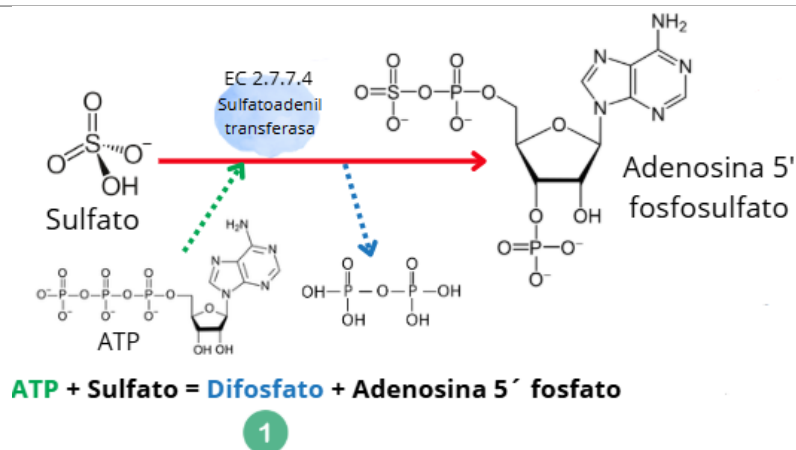
Tabla 12. *Reacción bioquímica de conversión de acetato a sulfuro descrito en KEGG.*

SULFURO A ACETATO		
Enzimas	Reacción	Fuente
Cisteína sintasa	O-acetil-L-serina + sulfuro de hidrógeno $\rightleftharpoons$ L-cisteína + acetato	KEGG (10)*

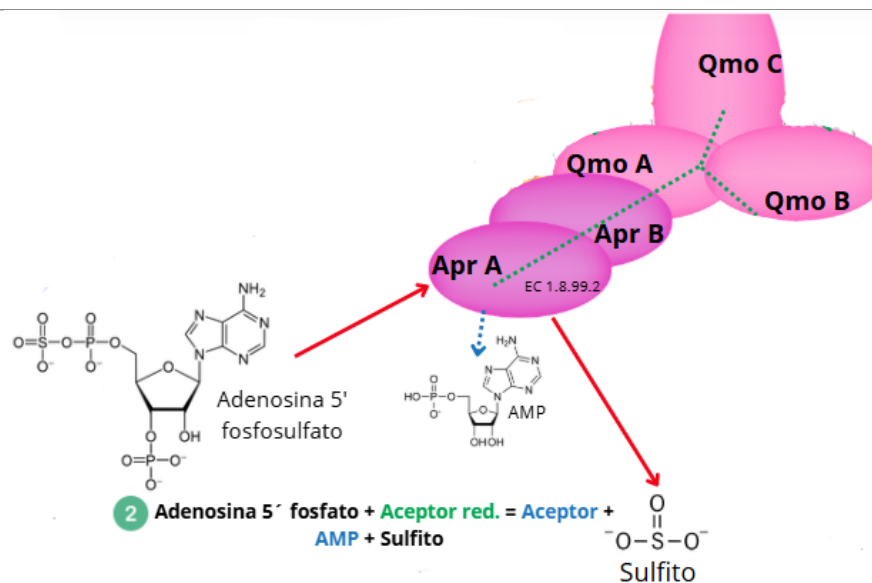
(10)* <https://www.genome.jp/entry/2.5.1.47>

Tabla 13. Descripción de la vía de sulfato a sulfuro.

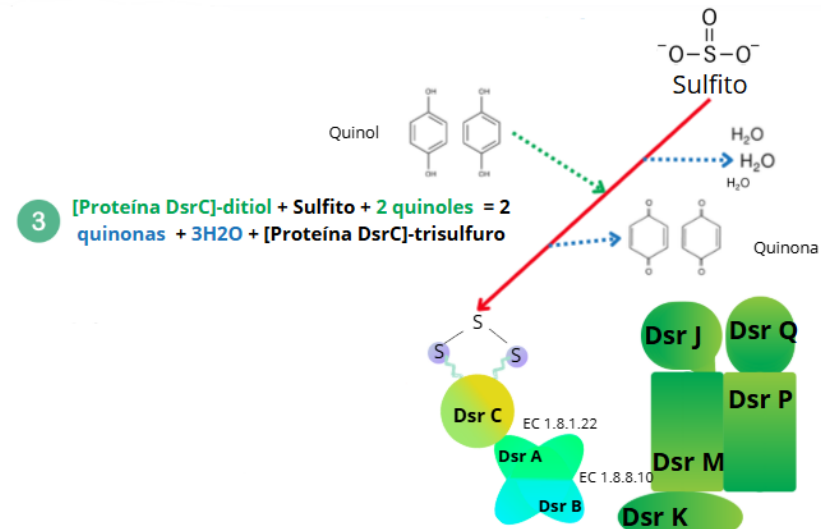
Activación del sulfato: La primera reducción de sulfato a sulfito requiere dos equivalentes de ATP, catalizados por la adeniltransferasa (ATP sulfurilasa), produciendo adenosina-5'-fosfosulfato (APS).



Reducción de APS: El APS es reducido a sulfito y AMP mediante la APS reductasa, una enzima que aumenta el potencial redox garantizando la reducción por electrones derivados del hidrógeno. El complejo AprAB está acoplado al transporte electrónico.



El módulo **DsrJOP** contiene **DsrP**, una subunidad de membrana integral con diez hélices transmembrana predichas, y **DsrO**, una subunidad de hierro-azufre predicha para coordinar de tres a cuatro grupos $[4\text{Fe-4S}]^{2+/1+}$, que juntos se parecen al módulo **NrfCD** interactuante con quinona ampliamente difundido en las oxidorreductasas bacterianas. Además, las proteínas **DsrMK** están presentes en todos los organismos que contienen **DsrAB** y **DsrC**, este módulo cataliza la reducción de **DsrC**-trisulfuro a sulfuro de hidrógeno a través de la oxidación del grupo de quinonas (Barbosa et al., 2024).



Los electrones del grupo de quinonas reducen consecutivamente los hemo b de **DsrM**, el centro catalítico no cubano $[4\text{Fe-4S}]^{3+/2+}$ de **DsrK** y, finalmente, el **DsrC**-trisulfuro, con liberación concomitante de sulfuro (Barbosa et al., 2024).

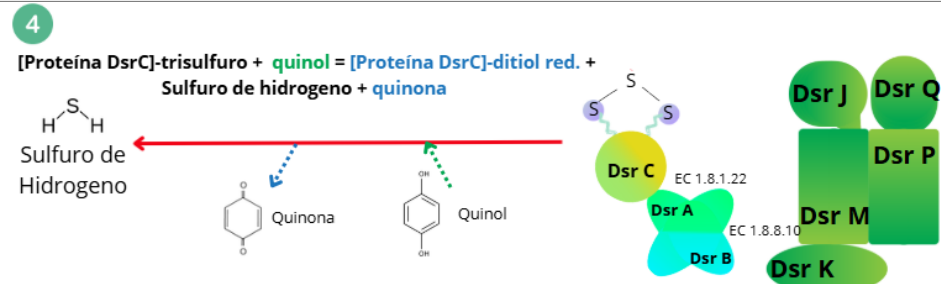
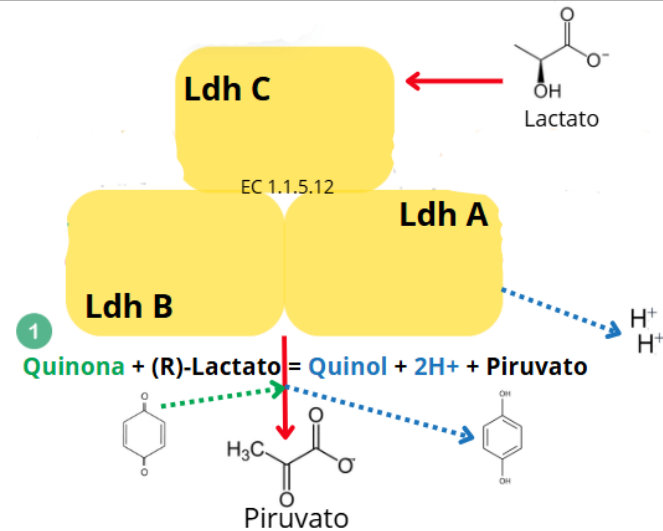


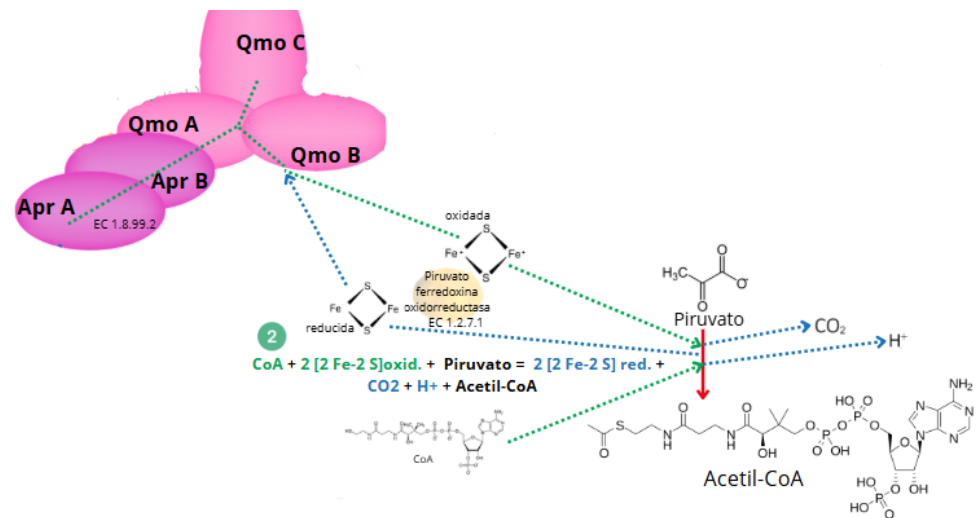
Tabla 14. Descripción de la vía de lactato a acetato.

Oxidación de lactato: La enzima lactato deshidrogenasa independiente de nucleótidos de pirimidina, ligada a la membrana y orientada al citoplasma, transforma el lactato en piruvato.

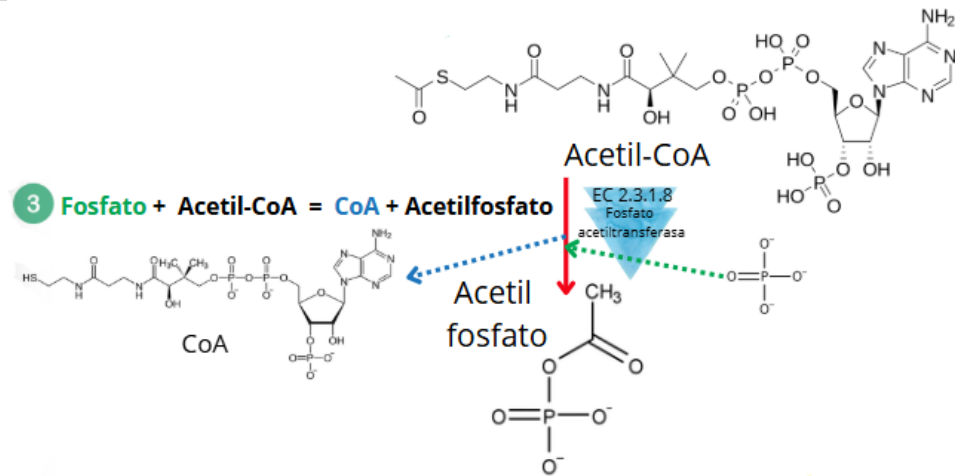


Oxidación de piruvato: El piruvato es oxidado a Acetil-CoA y CO₂ mediante la piruvato:ferredoxina oxidorreductasa (PFOR) soluble.

La **PFOR** cumple con 3 funciones simultáneamente, producir acetil-CoA, reducir ferredoxina (portador de electrones bajo potencial redox) y convierte la fuente de carbono del lactato a piruvato en energía utilizable en forma de electrones.



El acetil-CoA se convierte en acetilfosfato mediante la enzima fosfato-acetiltransferasa



El acetato quinasa transforma el acetilfosfato en acetato, generando ATP.

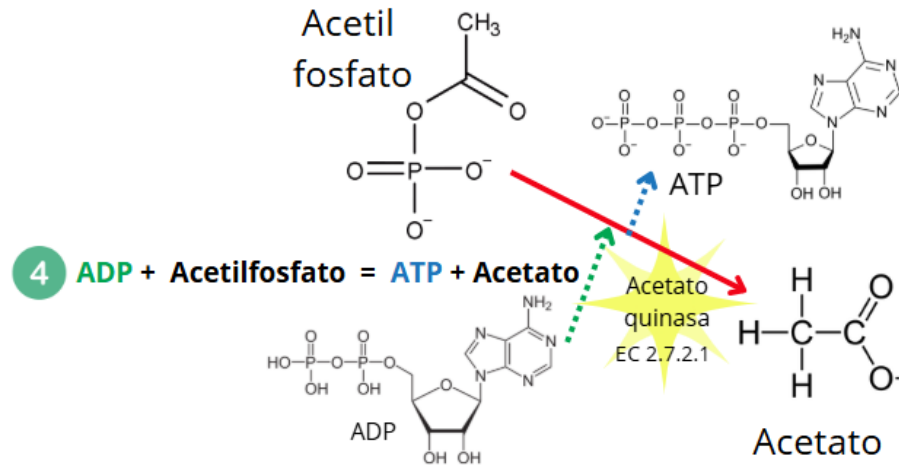
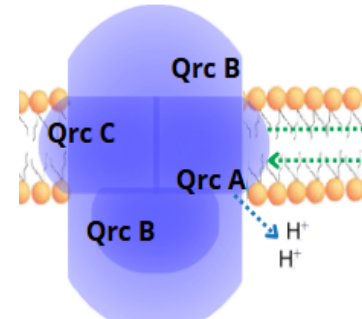


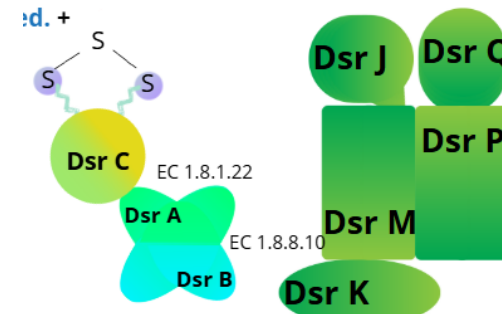
Tabla 15. Complejos transmembrana conservados (TMC) identificados en las rutas metabólicas del lactato y sulfato.

QmoABC: Fundamental para la reducción de sulfato, proporciona electrones a la APS reductasa.

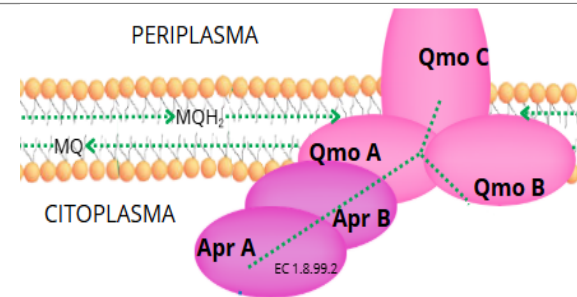


DsrMKJOP: El Dsr-trisulfuro interactúa fuertemente, como posible sustrato, con la DsrMKJOP (unido a membrana y al flujo de electrones).

La constante de Michaelis-Menten (K_M) para el DsrC-trisulfuro probablemente esté en el rango 200 ± 27 nM a 25°C (Barbosa et al., 2024).



QrcABCD: Presente en algunos géneros de *Desulfovibrio*, que recibe electrones de deshidrogenasas periplásmicas y de la deshidrogenasa de formiato a través del citocromo c_3 tetrahemo tipo I (TpI c_3) (Keller, 2014).



Mediante el análisis integrado de bases de datos bioinformáticas y literatura especializada, se estableció la ruta metabólica de *O. alaskensis*, la cual se presenta en la Figura 17.

La ruta metabólica de *Oleidesulfovibrio alaskensis* analizada en este trabajo (resumida en la figura 17) consta de dos vías principales interconectadas:

A) Vía de Reducción del Sulfato.

B) Vía de Oxidación del Lactato

Las cuales están acopladas a través del flujo de electrones y la conservación de energía.

En la vía del sulfato, el proceso inicia con la activación del sulfato (SO_4^{2-}), una molécula químicamente inerte, mediante la enzima ATP sulfurilasa (también denominada sulfato adeniltransferasa). En esta reacción el sulfato reacciona con ATP para formar adenosina-5'-fosfosulfato (APS) y pirofosfato (PPi). Posteriormente, el APS es reducido a sulfito (SO_3^{2-}) por la enzima APS reductasa, utilizando electrones provenientes del sistema de transporte electrónico, generando AMP como producto. El sulfito es posteriormente reducido a sulfuro mediante el sistema disimilatorio de reducción de sulfito. En esta etapa participa el complejo disimilatoria sulfito reductasa (complejo DsrAB). Este complejo cataliza la reacción del sulfito utilizando su cofactor catalítico siroheme-[4Fe-4S], generando especies reducidas de azufre que son transferidas a la proteína DsrC. Como resultado, DsrC alterna entre su forma reducida (ditiol) y un estado oxidado en la forma de trisulfuro (DsrC-trisulfuro), un intermediario clave en esta vía. Posteriormente, el DsrC-trisulfuro es reducido a sulfuro de hidrógeno (H_2S) por el complejo de membrana DsrMKJOP. Dentro de este sistema, el módulo DsrMK cataliza directamente la reducción del trisulfuro, mientras que el subcomplejo DsrJOP participa en la transferencia electrónica desde el pool de quinonas. En particular, DsrP corresponde a una proteína integral de membrana con múltiples hélices transmembrana, mientras que DsrO contiene centros hierro-azufre ([4Fe-4S]) que facilitan el flujo electrónico. Este proceso está acoplado a la oxidación de menaquinol (MQH_2) a menaquinona (MQ), lo que permite la transferencia de electrones hacia el DsrC-trisulfuro. En consecuencia, se libera H_2S y se regenera la forma ditiol de DsrC (Barbosa et al., 2024). De manera simultánea, este flujo electrónico contribuye a la generación de un gradiente electrónico a través de la membrana, el cual es aprovechado para la síntesis de ATP en la vía del lactato.

En la vía del lactato, el lactato actúa como principal donador de electrones y fuente de carbono. Este es oxidado a piruvato por una lactato deshidrogenasa de membrana (LdhABC),

independiente de NAD^+/NADH soluble y acoplada al pool de quinonas (menaquinonas), generando quinoles (menaquinoles) reducidos. Posteriormente, el piruvato es convertido a acetil-CoA por la enzima piruvato ferredoxina oxidoreductasa, produciendo CO_2 y reduciendo ferredoxina. El acetil-CoA es transformado en acetilfosfato por la fosfato acetiltransferasa (Pta), y posteriormente en acetato por la acetato quinasa (Ack), generando ATP por fosforilación a nivel de sustrato. Esta etapa representa una de las principales fuentes de energía metabólica.

El acoplamiento entre ambas vías ocurre a través del transporte de electrones. El complejo QmoABC transfiere electrones desde el pool de menaquinoles hacia la APS reductasa (AprAB), facilitando la reducción de APS. Asimismo, otros donadores de electrones como hidrogenasa periplásmicas y la formiato deshidrogenasa contribuyen al flujo electrónico a través de citocromos tipo c, incluyendo el citocromo c_3 tetrahemo.

Finalmente, la generación de biomasa en *O. alaskensis* está estrechamente asociada a la oxidación de lactato, ya que esta vía no solo proporciona energía (ATP), sino también intermediarios centrales (como la Acetil-CoA) necesarios para la biosíntesis celular, en concordancia con lo reportado para bacterias sulfato-reductoras.

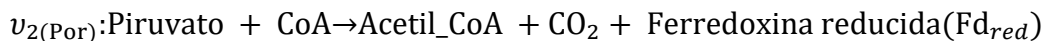
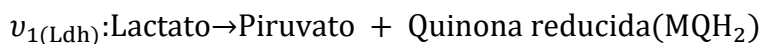
4.2.3. *Análisis de grados libertad para la ruta metabólica propuesta*

Para calcular numéricamente los GL de la ruta metabólica propuesta (Fig. 17), se siguió la metodología que considera como primer paso proponer un sistema lineal de ecuaciones estequiométricas independientes, contabilizando sus flujos y nodos principales, para finalmente comparar los resultados sobre sistemas sulfato-reductoras (*Desulfovibrio* spp.).

a) Cálculo de Flujos (F), Nodos (N) y Mediciones (M)

Para que el modelo tenga sentido matemático, se extrae la red metabólica central simplificada de la Figura 17 sin considerar protones libres y agua por balance hidrostático:

A. Los Flujos del Sistema ($F = 11$)



$v_{4(\text{Ack})}$: Acetilfosfato + ADP $\rightarrow$ Acetato + ATP

$v_{5(\text{Sat})}$: Sulfato + ATP $\rightarrow$ APS + PPi

$v_{6(\text{Apr})}$: APS + Aceptor_{red} $\rightarrow$ Sulfito + AMP + Aceptor_{ox}

$v_{7(\text{Dsr AB})}$: Sulfito + 2Quinoles $\rightarrow$ Proteína DsrC_{trisulfuro}

$v_{8(\text{Dsr C/K/J/M/O})}$:

Proteína DsrC_{trisulfuro} + Quinol $\rightarrow$ Sulfuro de Hidrógeno + Quinona

$v_{9(\text{Qmo})}$: Reoxidación de quinonas acoplada al complejo Apr

$v_{10(\text{Qrc})}$: Complejo aceptor de electrones/bombeo de protones transmembranal

$v_{11(\text{Biomasa})}$: Consumo de ATP y Acetil – CoA

B. Los Nodos Internos en Estado Estacionario ($N = 8$). Metabolitos que están obligados a balancearse netamente en el citoplasma (Entradas = Salidas):

1. Piruvato
2. Acetil-CoA
3. Acetilfosfato
4. APS
5. Sulfito
6. [Proteína DsrC]-trisulfuro
7. Pool Redox de Quinonas/Quinoles
8. Ferredoxina

C. Mediciones Extracelulares ($M = 6$)

1. Tasa de consumo de Lactato
2. Tasa de producción de Acetato
3. Tasa de consumo de Sulfato
4. Tasa de producción de Sulfuro
5. Tasa de producción de Biomasa
6. Tasa de producción CO_2

D. Cálculo de los Grados de Libertad (GL)

Se aplica la ecuación para el Análisis de Flujos Metabólicos (MFA)

$$GL = 11 - 8 - 6 = -3$$

Desde la perspectiva de Análisis de Flujos Metabólicos (MFA), un valor de $GL = -3$ clasifica a esta red como un sistema matemáticamente sobredeterminado. Biológicamente, esto significa que el modelo posee un índice de redundancia igual a 3, lo que implica que se cuentan con restricciones analíticas independientes adicionales a las estrictamente necesarias para resolver la topología interna del mapa metabólico. Desde el punto de vista de práctico, tener redundancia es una ventaja ya que sirve como un Filtro de "Caja Negra" contra mediciones defectuosas por ejemplo si existiera un sensor de lactato y tuviera un error del 20% en un sistema $GL = 0$ no se detectaría y los flujos calculados de ATP estarían mal y existiría un residual lo cual no cuadraría al balance de carbono. Lo más valioso del mapa son los complejos que acoplan la energía (la translocación de protones y la ATP sintasa). Como no se puede instrumentar un sensor dentro de la bacteria para medir cuánta energía genera cada complejo en tiempo real, las 3 dimensiones de redundancia exterior "acorralan" matemáticamente a las variables internas, obligándolas a tomar el único valor físico y termodinámico real posible. Es decir, al medir 6 variables, se crearon 3 dimensiones de control independientes (Carbono, Azufre y Electrones). Estas 3 dimensiones actúan como paredes matemáticas que acorralan a los flujos internos de la membrana, obligándolos a tomar un único valor real y permitiéndonos corregir los errores del laboratorio con rigor estadístico. La configuración experimental diseñada (6 mediciones extracelulares) transforma el mapa metabólico teórico en una poderosa herramienta de diagnóstico cuantitativo. Pasar de un modelo puramente teórico ($GL=0$) a un modelo sobredeterminado ($GL=-3$) permite discriminar con precisión matemática el flujo real de electrones y la tasa de síntesis de ATP en condiciones anaerobias reales. El cálculo de $GL = -3$ mostró que la red catabólica central de las bacterias sulfato-reductoras está fuertemente acoplada. Esto coincide con los hallazgos cuantitativos de Tang et al. (2007), quienes mediante perfiles metabólicos asistidos por ^{13}C determinaron que la práctica totalidad del lactato sigue la vía lineal hacia acetato. Además, el balance perfecto en los nodos redox internos está sustentado por los modelos quimiosmóticos de Pereira et al. (2011) y Keller et al. (2014), donde los complejos transmembrana Qmo y Dsr operan como compuertas estequiométricas rígidas que ligan la oxidación del donador con la reducción del aceptor a través del pool de menaquinonas.

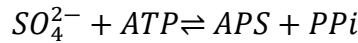
4.3 Modelo cinético estructurado de la ruta metabólica propuesta

4.3.1 Modelo estructurado propuesto para la ruta metabólica para operación por lote.

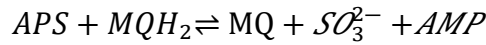
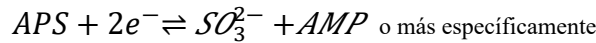
Las reacciones estequiométricas mejor identificadas en la literatura basada en Keller et al., (2014), López-Pérez (2013), Neria-González (s.f.) y las bases de datos bioinformáticas, que son la base para el modelado estructurado:

1) Activación y reducción de sulfato

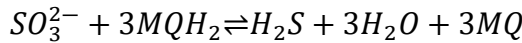
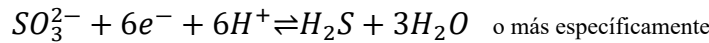
a) Activación del sulfato (ATP sulfurilasa)



b) Reducción de APS (APS reductasa, AprAB + QmoABC)

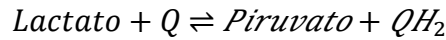


c) Reducción de sulfito a sulfuro (DsrAB + DsrC + DsrMKJOP)

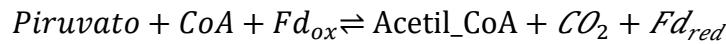


2) Oxidación de lactato (fuente de electrones y carbono)

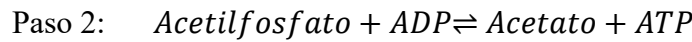
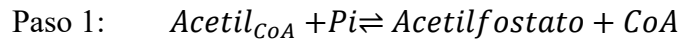
a) Lactato a piruvato (LdhABC, acoplada a quinonas)



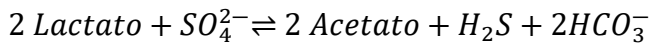
b) Piruvato a acetil-CoA (PFOR)



c) Formación de acetato (generación de ATP en 2 pasos)



3) Reacción global simplificada



Modelo propuesto:

Balance para la concentración de sulfato (S):

$$\frac{dS}{dt} = -Y \cdot \mu_{max} \left[1 - \left(\frac{P}{S_1} \right) \right]^a \left[\frac{S}{K_S + S} \right] X \cdot L^b \quad (7)$$

Balance para la concentración de adenosina 5' fosfosulfato (B):

$$\frac{dB}{dt} = \mu_{max1} \left(\frac{B \cdot S}{k_b + B} \right) X \left(\frac{C}{K_{co} + C} \right) \quad (8)$$

Balance para la concentración de sulfito (D):

$$\frac{dD}{dt} = \mu_{max2} \left(\frac{D \cdot B}{k_d + D} \right) X \left(\frac{C}{K_{co} + C} \right) \quad (9)$$

Balance para la concentración de sulfuro (P):

$$\frac{dP}{dt} = (\mu_{max} \cdot YP) \left[1 - \left(\frac{P}{S_1} \right) \right]^a \left[\frac{S}{k_s + S} \right] X \cdot L^b \quad (10)$$

Balance para la concentración de lactato (L):

$$\frac{dL}{dt} = -(\mu_{max3}) \left[\frac{k_{ace}}{A + k_{ace}} \right] \left[\left(\frac{L^c}{k_{lac} + L^c} \right) \right] X \quad (11)$$

Balance para la concentración de piruvato (E):

$$\frac{dE}{dt} = \mu_{max4} \left(\frac{E \cdot L}{k_e + E} \right) X \left(\frac{C}{K_{co} + C} \right) \quad (12)$$

Balance para la concentración de acetil-CoA (F):

$$\frac{dF}{dt} = \mu_{max6} \left(\frac{F \cdot E}{k_f + F} \right) X \left(\frac{C}{K_{co} + C} \right) \quad (13)$$

Balance para la concentración de acetil fosfato (G):

$$\frac{dG}{dt} = \mu_{max7} \left(\frac{G \cdot F}{k_g + G} \right) X \left(\frac{C}{K_{co} + C} \right) \quad (14)$$

Balance para la concentración de acetato (A):

$$\frac{dA}{dt} = (\mu_{max3}) \left[\frac{k_{ace}}{A + k_{ace}} \right] \left[\left(\frac{L^c}{k_{lac} + L^c} \right) \right] X \quad (15)$$

Balance para la concentración de dióxido de carbono (C):

$$\frac{dC}{dt} = (\mu_{max5}) \left[\frac{k_{co}}{C + k_{co}} \right] \left[\left(\frac{L^c}{k_{lac} + L^c} \right) \right] X \quad (16)$$

Balance para la concentración de biomasa (X):

$$\frac{dX}{dt} = (-\mu_{max}) \left[1 - \left(\frac{P}{S_1} \right) \right]^a \left[\left(\frac{L^c}{k_{lac} + L^c} \right) \right] X - k_d \cdot X \cdot L^b \quad (17)$$

Tabla 16. Variables del modelo estructurado.

Nombre de la variable	Nomenclatura en el modelo
Sulfato	S
Adenosina 5' fosfosulfato	B
Sulfito	D
Sulfuro de hidrógeno	P
Biomasa	X
Lactato	L
Piruvato	E
Acetil-CoA	F
Acetilfosfato	G
Acetato	A
CO₂	C

El modelo considera concentraciones iniciales de sulfato $[S_o] = 5655$ mg/L; sulfuro $[P_o] = 29$ mg/L; biomasa $[X_o] = 20$ mg/L; lactato $[L_o] = 4640$ mg/L; acetato $[A_o] = 0.1$ mg/L; Dióxido de carbono $[C_o] = 10$ mg/L; adenosina 5' fosfosulfato $[B_o] = 0.1$ mg/L; sulfito $[D_o] = 0.1$ mg/L; piruvato $[E_o] = 0.1$ mg/L; acetil-CoA $[F_o] = 0.01$ mg/L; acetilfosfato $[G_o] = 0.1$ mg/L. Así mismo, considera parámetros cinéticos de $SI = 680$; $k_{ace} = 1500$; $k_{lac} = 95000$; $k_s = 0.0009048$; $k_{co} = 900$; $k_d = 0.04$; $\mu_{max} = 0.13$; $\mu_{max_3} = 0.6$; $\mu_{max_5} = 0.12$; $Y = 7.5$; $YP = 1$; $a = 1.9$; $b = 0.001$; $c = 1.5$; $k_b = 0.036$; $\mu_{max_1} = 0.000056$; $k_{d2} = 0.12$; $\mu_{max_2} = 0.0055$; $k_e = 0.011$; $\mu_{max_4} = 0.000078$; $k_f = 0.017$; $\mu_{max_6} = 0.00015$; $k_g = 0.00012$; $\mu_{max_7} = 0.00038$. A partir de estos datos se obtuvieron los gráficos de la figura 18 que incluye sustratos, productos e intermediarios metabólicos.

4.3.2 Análisis de GL del modelo estructurado propuesto

El análisis de GL a partir de las ecuaciones diferenciales 7-17 del modelo matemático propuesto y su contraste con el mapa metabólico de la ruta requiere los siguientes supuestos:

- Definir los balances independientes en estado de equilibrio dinámico y las tasas de flujo que interactúan hacia el exterior.

- **Análisis de Variables y Ecuaciones del Modelo Matemático: Variables de Estado** (Concentraciones en el tiempo, $V = 11$ o $F=11$). El modelo plantea exactamente 11 ecuaciones diferenciales para 11 compuestos clave:
 1. S: Sulfato
 2. B: Adenosina 5'-fosfosulfato (APS)
 3. D: Sulfito
 4. P: Sulfuro
 5. L: Lactato
 6. E: Piruvato
 7. F: Acetil-CoA
 8. G: Acetilfosfato
 9. A: Acetato
 10. C: Dióxido de carbono
 11. X: Biomasa
- **Nodos Internos Restringidos ($N = 5$).** Para el análisis clásico de grados de libertad en estado estacionario celular $\frac{d_{\text{metabolito}}}{dt} = 0$, únicamente se consideran como nodos algebraicos aquellos compuestos puramente intermedios (que no escapan de forma neta al medio extracelular o acumulativo). De acuerdo a las ecuaciones cinéticas, los intermediarios metabólicos intramembranales/citoplasmáticos estrictos son:
 1. $B_{(\text{Adenosina 5'-fosfosulfato})} \rightarrow \text{Eq. (8)}$
 2. $D_{(\text{Sulfito})} \rightarrow \text{Eq. (9)}$
 3. $E_{(\text{Piruvato})} \rightarrow \text{Eq. (12)}$
 4. $F_{(\text{Acetil-CoA})} \rightarrow \text{Eq. (13)}$
 5. $G_{(\text{Acetil fosfato})} \rightarrow \text{Eq. (14)}$
- **Mediciones Extracelulares / Flujos Macroscópicos ($M = 6$).** Las variables restantes corresponden a compuestos que se consumen, secretan o acumulan de forma medible en el medio exterior (biorreactor). Las ecuaciones diferenciales asignan dinámicas macroscópicas directas acopladas a la concentración de Biomasa (X):
 1. Tasa de consumo de Sulfato (S) $\rightarrow \text{Eq. (7)}$
 2. Tasa de producción de Sulfuro (P) $\rightarrow \text{Eq. (10)}$
 3. Tasa de consumo de Lactato (L) $\rightarrow \text{Eq. (11)}$

4. Tasa de producción de Acetato (A) →Eq. (15)
 5. Tasa de producción de Dióxido de carbono (C) →Eq. (16)
 6. Tasa de crecimiento de Biomasa (X) →Eq. (17)
- Flujos / Pasos de Reacción Cinética (F = 11). Cada término de velocidad regulado por un parámetro $\mu_{\max,i}$ representa una velocidad de flujo metabólico neto que cruza el sistema. Contamos 11 velocidades principales acopladas a los términos cinéticos.
 - Cálculo del GL, sustituyendo los valores extraídos directamente de modelo propuesto:

$$GL = 11_{(\text{Velocidades cinéticas})} - 5_{(\text{Nodos internos})} - 6_{(\text{Flujos externos medidos})} = 0$$

GL=0, es estrictamente determinado, lo que ocurre porque el modelo matemático asume cinéticas acopladas rígidas, no existe redundancia libre independiente en el sistema de ecuaciones diferenciales, el modelo obliga a que las tasas internas se resuelvan de forma unívoca a partir de las trayectorias de las 6 variables medibles.³ Por lo tanto, si en el laboratorio se monitorea la velocidad con la que la bacteria consume Lactato, produce Acetato y consume Sulfato, se puede calcular matemáticamente el 100% de las velocidades de las reacciones internas (como la actividad de los complejos Qmo, Apr, Dsr y la síntesis exacta de ATP celular) mediante un sistema de ecuaciones lineales simple, sin necesidad de adivinar o usar optimizaciones complejas. El hecho de que el formalismo cinético resulte en GL= 0 demuestra que el modelo matemático propuesto en las ecuaciones (7-17) captura la continuidad de masa del mapa metabólico sin dejar "cabos sueltos" o rutas metabólicas alternativas no controladas. Sin embargo, aunque el sistema global posea cero grados de libertad para determinar el valor neto de los flujos en estado estacionario, la distribución temporal del transitorio depende estrictamente de la sensibilidad de estos parámetros cinéticos, impidiendo la acumulación tóxica de intermediarios como el sulfito (D) o el piruvato (E).

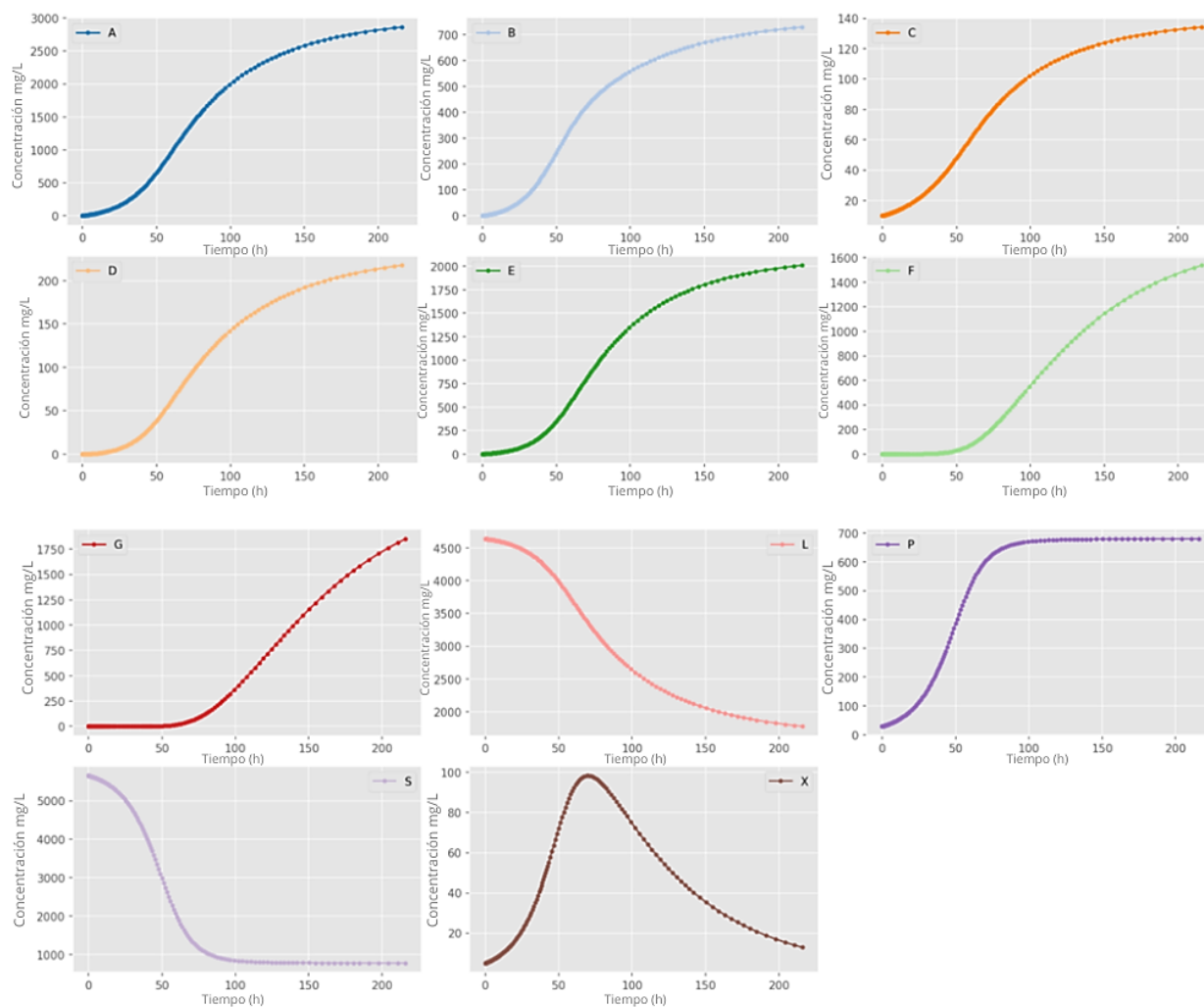


Figura 18. Cinética de *Oleidesulfovibrio alaskensis* de las concentraciones de lactato, acetato, sulfato. Sulfuro, biomasa y carbono, con enzimas involucradas de la ruta del lactato y sulfato.

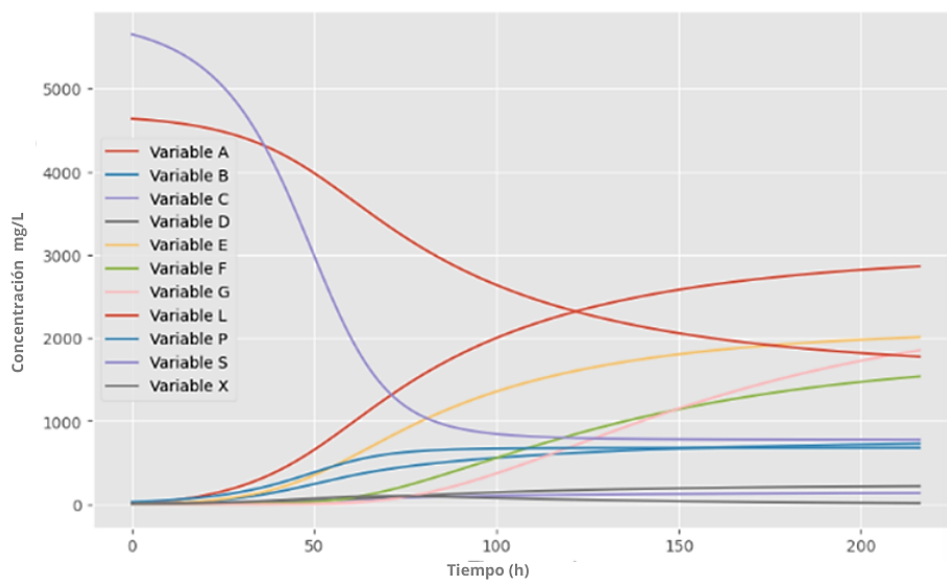


Figura 19. Concentraciones de todas las variables dinámicas del modelo estructurado.

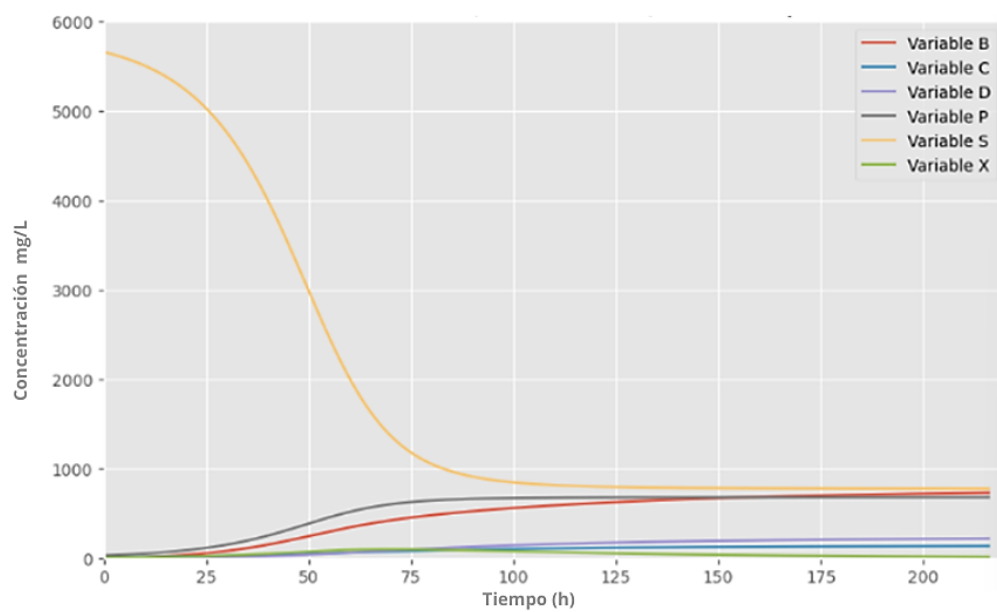


Figura 20. Concentraciones de las variables dinámicas de la vía del sulfato en el modelo estructurado.

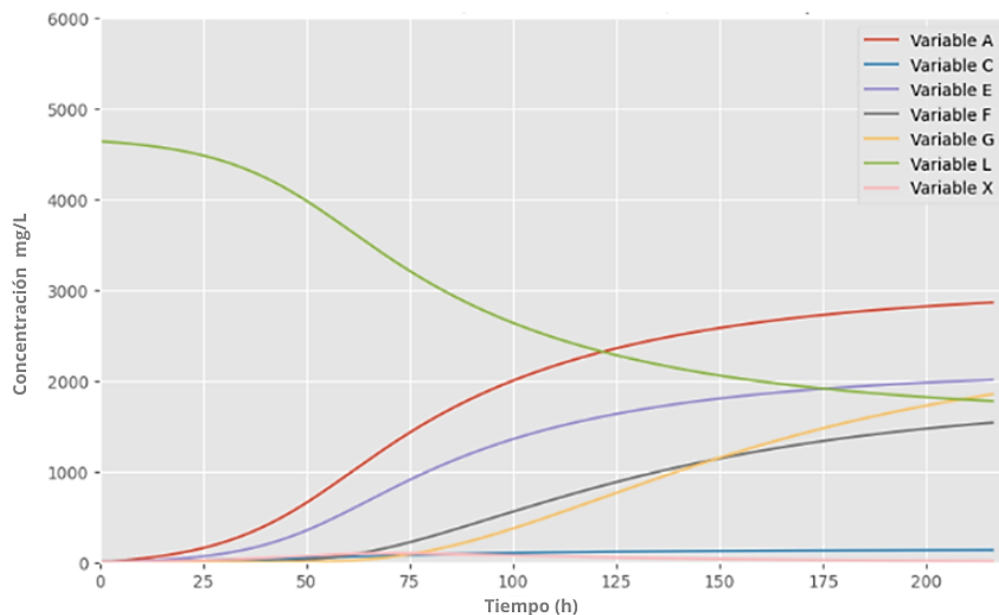


Figura 21. Concentraciones de las variables dinámicas de la vía del lactato en el modelo estructurado.

Las figuras 18 y 19 muestran los perfiles dinámicos de las concentraciones de los metabolitos y biomasa involucrados en la ruta metabólica de la reducción del sulfato acoplada a la oxidación del lactato. Se observa la evolución temporal del sulfato (S), adenosina-5'-fosfosulfato (B), sulfito (D), sulfuro de hidrógeno (P), biomasa (X), lactato (L), piruvato (E), acetyl-CoA (F), acetilfosfato (G), acetato (A) y dióxido de carbono (C). En la figura 20 se muestran solo las concentraciones asociadas a la vía del sulfato (S , D , P , C , B y X). En la figura 21 se muestran solo las concentraciones asociadas a la vía del lactato (A , C , E , F , G , L y X).

Se observa el consumo progresivo de los sustratos principales, lactato (L) y sulfato (S), acompañado de la formación de intermediarios metabólicos los cuales presentan dinámicas transitorias. Paralelamente, se evidencia la acumulación de productos finales como acetato (A), dióxido de carbono (C) y sulfuro de hidrógeno (P). La biomasa (X) muestra un incremento inicial seguido de una estabilización, reflejando la fase de crecimiento y posterior limitación por sustrato.

4.3.2.1 Sensibilidad

El análisis de sensibilidad para el modelo estructurado siguió la misma estructura que del modelo no estructurado obteniendo la tabla 16.

Tabla 17 Sensibilidad de los parámetros y las distintas variables.

	S	B	D	P	L	E	F	G	A	C	X
SI	<	<	>	>	<	<	<	<	<	<	<
K_{ace}	>	<	>	<	<	<	<	<	>	<	<
K_{lac}	<	<	<	>	>	<	<	<	>	<	>
K_s	>	<	<	<	>	<	<	<	<	<	<
K_{co}	<	<	<	<	<	<	<	<	<	>	>
K_d	>	<	>	>	>	>	>	>	>	<	<
μ_{max}	>	<	>	>	>	>	>	>	>	>	>
μ_{max_3}	>	<	>	<	>	<	<	>	>	<	<
μ_{max_5}	<	>	>	<	<	>	>	>	<	>	<
Y	>	<	<	>	<	>	>	>	>	<	>
YP	>	>	<	>	>	<	>	>	>	<	<
a	>	<	<	>	<	>	>	>	>	<	>
b	>	<	<	>	>	<	<	<	>	<	<
c	<	>	<	<	<	<	<	<	>	>	<
K_b	<	<	<	<	<	<	<	<	<	>	<
μ_{max_1}	<	>	>	<	<	<	<	<	<	<	<
K_{d2}	<	<	>	<	<	<	<	<	<	<	<
μ_{max_2}	<	<	>	<	<	<	<	<	<	<	<
K_e	<	<	<	<	<	>	<	<	<	<	<
μ_{max_4}	<	<	<	<	<	>	>	>	<	<	<
K_f	<	<	<	<	<	<	<	<	<	<	<
μ_{max_6}	<	<	<	<	<	<	>	>	<	<	<
K_g	<	<	<	<	<	<	<	>	<	<	<
μ_{max_7}	<	<	<	<	<	<	<	>	<	<	<

A_0	<	<	<	<	>	>	<	<	>	<	<
B_0	<	<	<	<	<	<	<	<	<	<	<
C_0	<	>	<	<	<	>	<	>	<	<	>
D_0	<	<	<	<	<	<	<	<	<	<	<
E_0	<	<	<	<	<	>	<	<	<	<	<
F_0	<	<	<	<	<	<	<	<	<	<	<
G_0	<	<	<	<	<	<	<	<	<	<	<
L_0	<	<	<	<	>	<	<	<	>	<	<
P_0	<	<	<	<	<	<	<	<	>	<	<
S_0	<	<	>	>	<	<	<	<	<	<	>
X_0	>	<	<	>	<	<	<	>	>	<	>

Criterios para medir la influencia en cada variable:

- Magnitud absoluta
- Normalizar valores
- Que no se cancelen entre sí y no generen efectos no lineales

Símbolo	Sensibilidad
>	Muy alta
>	Alta
<	Moderada
>	Débil
<	Despreciable

El análisis de sensibilidad arrojó un total de 385 figuras de parámetro y variable (figura A 32, Apéndice D). Los resultados se resumieron en la tabla 16 e indican que la tasa máxima de crecimiento (μ_{max}) es el parámetro con mayor influencia global, mostrando sensibilidad muy alta en múltiples variables del sistema, incluyendo sulfato (S), sulfuro (P), lactato (L), acetato (A) y biomasa (X). Asimismo, el rendimiento (Y) y (YP) presentan alta sensibilidad en varias variables, especialmente en productos y biomasa. Entre los parámetros cinéticos, ks , kd , a y algunos términos asociados a la inhibición muestran sensibilidad muy alta, pero de forma más localizada en metabolitos específicos. Además, las condiciones iniciales, particularmente X_0 , influyen fuertemente en varias variables, confirmando su rol determinante en la dinámica del bioproceso.

A partir de la tabla 17 se realizó una conversión a valores numéricos (material complementario) para obtener el mapa de calor realizado con Rstudio de la figura 22, donde se observa las relaciones entre parámetros y las relaciones entre variables.

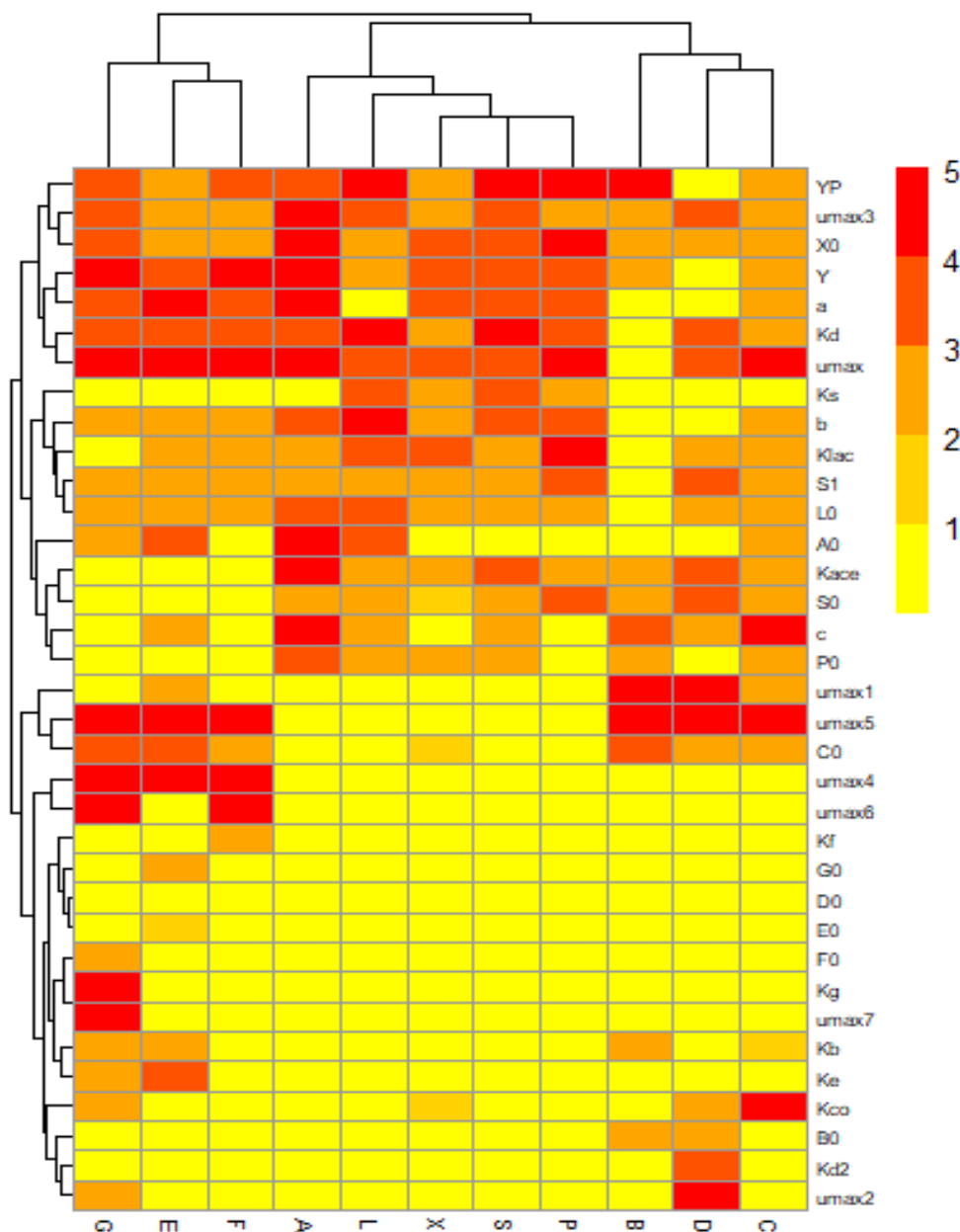


Figura 22. Mapa de calor de la sensibilidad del modelo estructurado, variables y parámetros.

El mapa de calor de sensibilidad (figura 22) revela una organización interna coherente, se identifica un subconjunto reducido de parámetros con influencia sistemática elevada sobre múltiples variables, como en análisis previos donde μ_{max} , Y , y YP siguen dominando, lo que sugiere

acoplamiento dinámico y posible colinealidad funcional. Así mismo, se observa que varios parámetros presentan impacto marginal. Además, los clustering de variables reflejan acoplamiento entre el crecimiento y producción, al igual que dependencia compartida de flujo metabólico, esto es coherente con sistemas metabólicos reales, donde pocas reacciones clave (enzimas limitantes) determinan el comportamiento global de la ruta.

Si comparamos el primer mapa de calor obtenido (figura 14) con este mapa de calor (figura 22), podemos concluir que el modelo completo no introduce ruido artificial, sino que amplifica una estructura ya presente en el núcleo metabólico.

Los parámetros enzimáticos como μ_{max_1} y μ_{max_4} resultaron tener una influencia baja en comparación con otros parámetros como μ_{max} . Esto no significa que los parámetros de esas enzimas no sean importantes, sino que su regulación podría estar subordinada a limitaciones más globales. Mori et al. (2017) demostraron en su estudio donde vincularon la tasa de crecimiento, la ingesta de la fuente de carbono y costos biosintéticos de *E. coli*, que el crecimiento óptimo es el resultado del equilibrio entre la maximización del rendimiento y la minimización de la carga proteica, esto sugiere que la baja sensibilidad de los parámetros individuales, comparada a la alta sensibilidad de la tasa máxima de crecimiento podría interpretarse bajo el marco de los costos de inversión enzimática. Es decir que la célula enfrenta una disyuntiva en la asignación de recursos limitados para la síntesis de proteínas, priorizando aquellas que maximizan la tasa de crecimiento global bajo las condiciones imperantes de inhibición por sulfuro, en lugar de optimizar rutas enzimáticas individuales.

4.3.2.2 *Análisis de trayectorias y rendimientos con PCA y Random Forest para la producción de H_2S (P).*

En esta sección se realizó un código que usa la simulación del modelo estructurado y su análisis de sensibilidad y se incorporó machine-learning para entender cómo los parámetros del modelo (velocidades, rendimientos, constantes) afectan la producción de H_2S (P), cómo influyen en la forma de las trayectorias temporales de las variables y que parámetros son más importantes para optimizar el proceso, usando librerías como `pyfoomb` (simulación de bioprocesos), `scikit-learn` (PCA, Random Forest) y `matplotlib` / `seaborn` (Visualización).

A partir de la clase `SequentialKinetic` obtenida de `BioprocessModel` de la librería de `pyfoomb` define un sistema de 11 ecuaciones diferenciales ordinarias (EDOs) que representan la

evolución temporal de las variables de estado (y): $A, B, C, D, E, F, G, L, P, S$ y X ; y los parámetros del modelo. Donde el método `rhs` calcula las derivadas de estado de las variables (`dXdt`, `dSdt`, etc) siguiendo cinéticas típicas de fermentación (crecimiento de biomasa con inhibición de producto y limitación de sustrato, muerte celular dependiente del sustrato, consumo de lactato y los pasos secuenciales de enzima a enzima en el metabolismo del microorganismo).

Se generó un dataset variando parámetros con la función `generate_bioprocess_dataset(n_samples=200)` donde toma los parámetros base y los varia aleatoriamente entre $\pm 30\%$ (factor 0.7 a 1.3) y para cada conjunto ejecuta una simulación completa, guarda los valores finales de todas las variables de estado, clasifica el resultado en una categoría (alto/medio/bajo rendimiento o rendimiento pobre) según los valores de `P_final` y `X_final` y retorna (x) una matriz de características (los parámetros variados), (y) etiquetas de clase y (`feature_names`) nombres de parámetros. Esto convierte el sistema en un dataset de aprendizaje supervisado.

En el análisis exploratorio y PCA para un subconjunto de parámetros, se muestra un pairplot de seaborn que colorea por clase y ayuda a visualizar relaciones entre pares de parámetros y separabilidad de las distintas categorías de rendimiento.

Después se realizó un análisis más completo con 3 etapas:

- 1) **Relación de parámetros con la formación de H_2S (P).** Aquí se calculan correlaciones lineales entre cada parámetro y `P_final`. Entrena un `Random Forest Regressor` para predecir `P_final` a partir de los parámetros. Obtiene la importancia de cada parámetro en la predicción. Y visualiza los 3 parámetros más importantes vs `P_final`.
- 2) **Relación de parámetros con la forma de trayectorias.** Se extraen características de la trayectoria de P (valor máximo, área bajo la curva, tasa máxima de crecimiento, etc.), se aplica `PCA` sobre esas características para reducir dimensionalidad. Usa `K-means` (3 clouster) para agrupar trayectorias según la forma. Entrena un `Random Forest Classifier` para predecir el clouster de trayectoria a partir de los parámetros. Por último, identifica que parámetros discriminan mejor entre formas de trayectoria.
- 3) **Comparación integrada.** Compara la importancia de cada parámetro para predecir `P_final` vs predecir la importancia de la trayectoria y destaca parámetros que afectan solo el valor final y no la forma, o viceversa.

La importancia de características (Random Forest) y las correlaciones revelan que parámetros controlan el resultado final y cuales afectan la cinética (forma de la curva). Además, ayuda a visualizar patrones complejos en las trayectorias o en los parámetros.

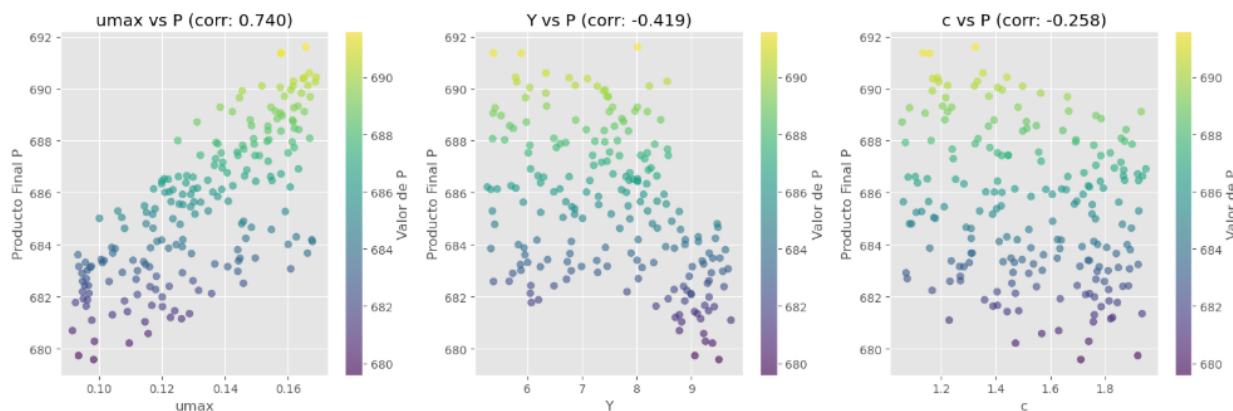


Figura 23. Análisis de correlación entre parámetros del modelo expandido y el producto (H_2S).

La figura 23 representa el análisis de correlación entre parámetros del modelo y el producto final P (H_2S), complementado con diagramas de dispersión para los parámetros más influyentes. El análisis de correlación muestra que μ_{max} presenta la mayor asociación positiva con P ($r=0.74$), indicando que los incrementos en la tasa máxima de crecimiento se traducen en mayores concentraciones finales de producto. Por otro lado, el rendimiento Y muestra una correlación negativa moderada de -0.42 , lo que sugiere un efecto compensatorio entre el crecimiento y la formación de producto. De igual manera, el parámetro c muestra una correlación negativa, aunque con menor magnitud ($r=-0.26$). El resto de los parámetros presentan correlaciones débiles menores a 0.2, indicando una influencia limitada sobre el producto en el rango evaluado.

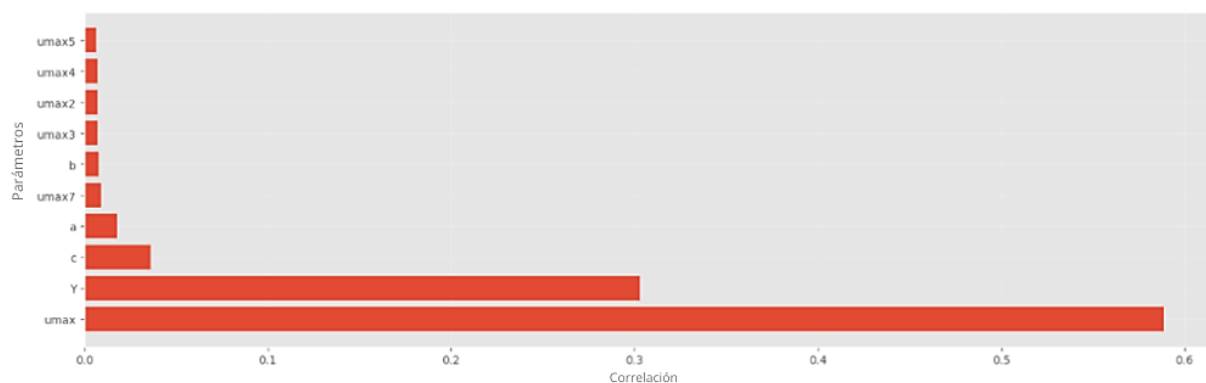


Figura 24. Ranking de los diez parámetros más relevantes en la predicción de formación de producto.

La figura 24 presenta un ranking de los diez parámetros más importantes para la predicción del producto según su contribución relativa dentro del modelo. Se muestra una dominancia significativa del parámetro μ_{max} , con una importancia aproximada de 0.59, es decir, más de la mitad de la contribución total, seguida por Y con una importancia cercana a 0.30, mientras que el resto de los parámetros presentan contribuciones significativamente menores a 0.05. Desde un punto de vista para la calibración del modelo, los ajustes en μ_{max} y Y resultan importantes para mejorar la capacidad predictiva del producto H_2S (P).

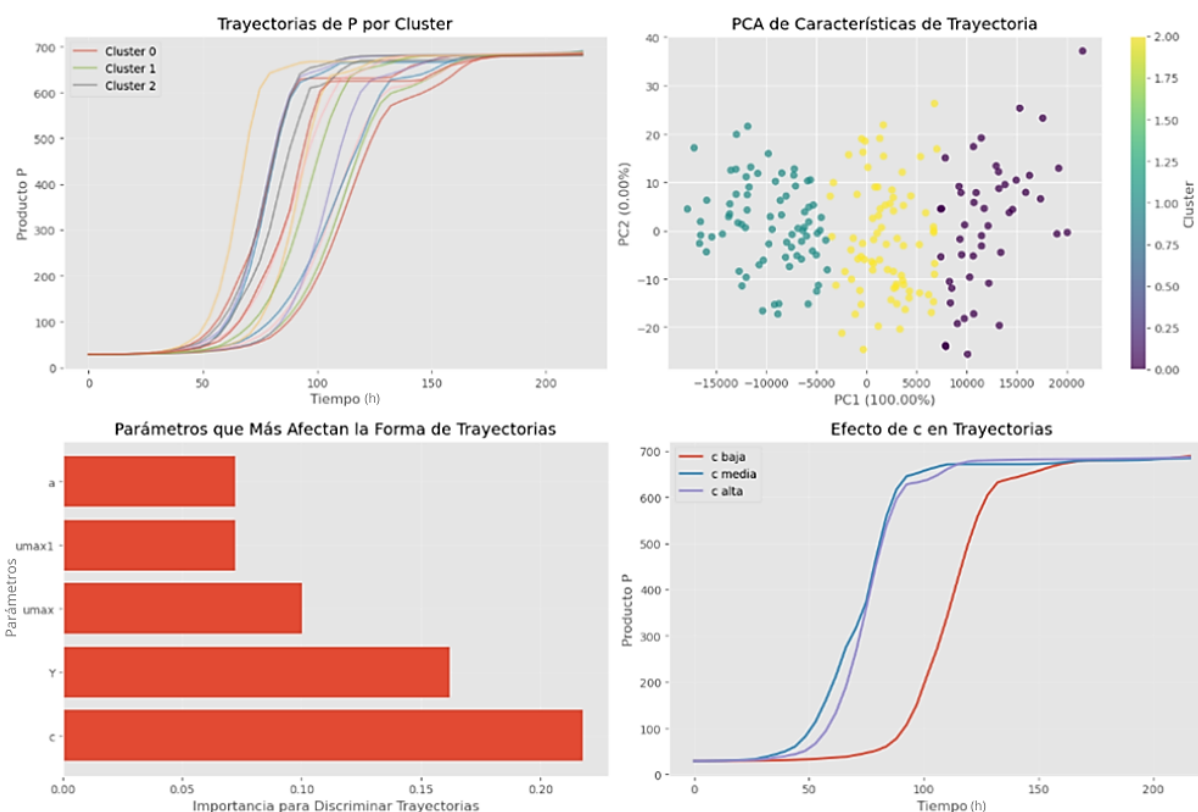


Figura 25. Análisis de las trayectorias dinámicas del producto en relación a los parámetros del modelo expandido.

La figura 25 presenta el análisis de la forma de las trayectorias dinámicas del producto (P) mediante reducción de dimensionalidad (PCA) y agrupamiento no supervisado. El análisis de componentes principales muestra que los dos primeros componentes explican prácticamente la totalidad de la varianza de las trayectorias indicando que la variabilidad dinámica del sistema puede representarse en un espacio de baja dimensionalidad.

La figura 25 deja ver la existencia de tres patrones dinámicos diferenciados, asociados principalmente a desplazamientos en el tiempo de crecimiento y velocidad de saturación de producto, más que a cambios en el valor final alcanzado. El PCA confirma esta separación de clústeres, mostrando agrupamientos bien definidos.

El análisis de importancia paramétrica para la forma de trayectorias indica que c y Y son los principales determinantes de la morfología de la dinámica, seguidos por μ_{max} y μ_{max_1} , mientras que a presenta una contribución menor. El análisis específico del efecto de c muestra que variaciones en estos parámetros modifican principalmente la pendiente y el tiempo de transición de la fase exponencial, desplazando la cinética de saturación por producto sin alterar significativamente el estado estacionario final.

Tabla 18. Comparación de las correlaciones de importancia para la concentración de la producción de H_2S y la trayectoria.

Parámetro	Correlación de concentración	Correlación de trayectoria	Diferencia
μ_{max}	0.588464	0.100202	0.488262
Y	0.303091	0.162148	0.140943
c	0.036272	0.217575	0.181303
a	0.017951	0.072229	0.054278
μ_{max_7}	0.008817	0.046399	0.037583
b	0.007760	0.047105	0.039345
μ_{max_3}	0.007362	0.054133	0.046771
μ_{max_2}	0.007278	0.56816	0.049538

Fuente: elaboración propia.

El parámetro más importante para la trayectoria, pero no para la formación de producto es c con un valor de importancia en la concentración de 0.036 y un valor de importancia en la trayectoria de 0.218.

4.3.2.3 *Análisis de Random Forest, sensibilidad y plausibilidad metabólica en el modelo estructurado.*

Se realizó un análisis de sensibilidad global para determinar qué parámetros del modelo afectan más a la producción de H_2S con la finalidad de saber qué parámetros medir o ajustar con más cuidado para predecir o controlar el bioproceso.

A partir de la clase `SequentialKinetic` obtenida de `BioprocessModel` de la librería de `pyfoomb` define un sistema de 11 ecuaciones diferenciales ordinarias (EDOs) que representan la evolución temporal de las variables de estado (y): $A, B, C, D, E, F, G, L, P, S$ y X ; y los parámetros del modelo. Donde el método `rhs` calcula las derivadas de estado de las variables (`dXdT`, `dSdt`, etc) siguiendo cinéticas típicas de fermentación (crecimiento de biomasa con inhibición de producto y limitación de sustrato, muerte celular dependiente del sustrato, consumo de lactato y los pasos secuenciales de enzima a enzima en el metabolismo del microorganismo). Además, se incluyeron protecciones para evitar valores negativos o divisiones por 0 para el análisis del modelo.

Se usó machine-learning para un análisis de sensibilidad con Random Forest con el objetivo de variar sistemáticamente los parámetros del modelo, ejecutar muchas simulaciones y ver que parámetros explican más la variabilidad en los resultados donde:

- 1) Se generó el dataset (`generate_complete_dataset`). Se toman los parámetros que se van a variar, por cada simulación (por defecto 100 o 200) se multiplica cada parámetro por un factor aleatorio entre 0.8 y 1.2 (variación del $\pm 20\%$). Se ejecuta el modelo con parámetros aleatorios y se guardan los valores finales de cada variable de estado (por ejemplo, `P_final`, `X_final`) y opcionalmente, toda la trayectoria y características como valor máximo, área bajo la curva etc. Esto genera una matriz x (muestras x parámetros) y vectores objetivo (por ejemplo, `P_final` para análisis de regresión, o una clasificación si se encuentra en “bajo”, “medio” y “alto” rendimiento).
- 2) Se realizó un análisis de clasificación (`analyze_parameter_importance`). Aquí se convierte la producción final en categorías de rendimiento (bajo/medio/alto) mediante `LabelEncoder`. Se entrena un `RandomForestClassifier` con los datos (80% entrenamiento, 20% prueba). Se evalúa precisión y se reporta matriz de confusión / classification report. Por último, se extrae `feature_importances_`, que indica qué parámetros son más importantes para predecir la categoría de rendimiento.

- 3) Se realizó un análisis de regresión (`analyze_numeric_sensitivity`). Es similar al anterior, pero usando `RandomForestRegressor` para predecir el valor numérico de **P** (producción de H_2S), el resultado es una lista de importancias de parámetros. Al comparar ambos métodos (clasificación vs regresión) se puede verificar la consistencia de los resultados. Si el top 5 de parámetros importantes coincide entre ambos, se tiene mayor confianza.
- 4) Se realizó un análisis multivariado para todas las variables (`analyze_all_variables_sensitivity`). Se repite el proceso de regresión para cada variable de estado (**A**, **B**, **C**, **D**, **E**, **F**, **G**, **L**, **P**, **S** y **X**). Genera un mapa de calor que muestra como la importancia de cada parámetro varía según la variable de salida. Calcula la importancia global (suma de importancias sobre las variables) y la especificidad (qué parámetros afectan pocas variables específicas y su correlación).
- 5) Se realizó un análisis de trayectorias (`analyze_trajectory_sensitivity`). Aquí en lugar de usar solo el valor final, se extrae características dinámicas de las trayectorias (valor máximo, tiempo al pico, área bajo la curva, tasa máxima de cambio). Para cada característica se entrena un Random Forest y se obtienen importancias. Esto muestra que parámetros controlan, por ejemplo, la velocidad de crecimiento o la forma de la curva del producto **P** (H_2S).

Este código genera varios gráficos (con matplotlib y seaborn):

- 1) Mapa de calor de correlaciones.
- 2) Gráfico de barras con el R^2 obtenido para cada variable (qué tan bien predice el modelo de Random Forest esa variable a partir de los parámetros).
- 3) Importancia global de los parámetros (top 10).
- 4) Especificidad, parámetros que son muy importantes solo para una variable.

Al final del código se agregó un apartado de interpretación metabólica a partir de las funciones de (`check_biological_plausibility` y `generate_comprehensive_report`) que comparan si los parámetros más importantes (por ejemplo, μ_{max} , **Y**, **a**) tienen sentido metabólico. De esta manera se hace un reporte final con parámetros universales (parámetros que afectan muchas variables), parámetros específicos para variables de interés (ejemplo, μ_{max3} para lactato) y una interpretación metabólica corta basada en la literatura de Neria-González et al (2010) y López-Pérez (2013).

La interpretación metabólica se define a partir de un conjunto de parámetros que tienen un rol metabólico conocido en el proceso de fermentación:

- `umax` : tasa específica de crecimiento microbiano (h^{-1}) en condiciones específicas.
- `Y` : rendimiento de biomasa a partir de sustrato (g biomasa / g sustrato).
- `a` : exponente de inhibición por producto. Controla como la acumulación del producto P (H_2S) frena el crecimiento celular.
- `b` : exponente del efecto del sustrato en la muerte celular.
- `ks` : constante de afinidad por sustrato.
- `c` : exponente de cinética de lactato. Modula como el lactato afecta los pasos posteriores.

La función `check_biological_plausibility(importance_df)` compara los parámetros que el machine-learning considera importantes (top 5) con una lista predefinida de parámetros que son importantes metabólicamente (por ejemplo, `umax`, `Y`, `a`, `YP`, `ks`, etc.) y si al menos 2 de los 5 parámetros permanecen en esa lista, se dice que el análisis es metabólicamente plausible. En caso contrario podría ser indicativo de que el modelo o los datos tiene algún problema (como puede ser ruido excesivo o algún ruido en la simulación). Dentro de `verify_sensitivity_results` y `generate_comprehensive_report` se incluye un diccionario `bio_interpretations` que asigna a cada parámetro clave una frase con su papel metabólico para P :

- `umax` : “Tasa máxima de crecimiento microbiano – afecta directamente la productividad”.
- `a` : “Exponente de inhibición por producto – controla como el producto limita el crecimiento”.
- `b` : “Exponente de efecto del sustrato en la muerte celular – influye en la longevidad del cultivo”.
- `umax3` : “Consumo de lactato – clave para metabolitos posteriores”.
- `Y` : “Rendimiento – crucial para sustrato y biomasa”.

El código imprime estas recomendaciones finales para que el usuario comprenda el paso metabólico que debe controlar para mejorar la fermentación. Además, da una lista que conecta cada parámetro importante con la variable metabólica que más afecta. Si c (exponente de lactato) aparece como altamente importante para la predicción de L (lactato) y P (sulfuro), se interpreta que la sensibilidad del consumo de lactato es crucial para el metabolismo. Si kd (tasa de muerte

celular) es importante para X (biomasa) pero no para P se infiere que la muerte afecta la biomasa, pero no el producto directamente, quizás porque el producto es acumulado también por células no viables.

Cabe mencionar que el mapa de calor (`importance_matrix`) permite ver que parámetros son específicos de una variable (ejemplo, k_{ace} para A) y cuáles son globales (μ_{max} que afecta a casi todas). Al final, esta interpretación metabólica sirve para recomendar experimentos o estrategias de control, por ejemplo: en parámetros universales se deben calibrar primero en experimentos independientes, en parámetros específicos se necesita diseñar ensayos enfocados en la ruta metabólica (como sería medir las enzimas involucradas) y si el análisis muestra que la inhibición por producto (a) es determinante, se sugiere eliminar el producto de forma continua.

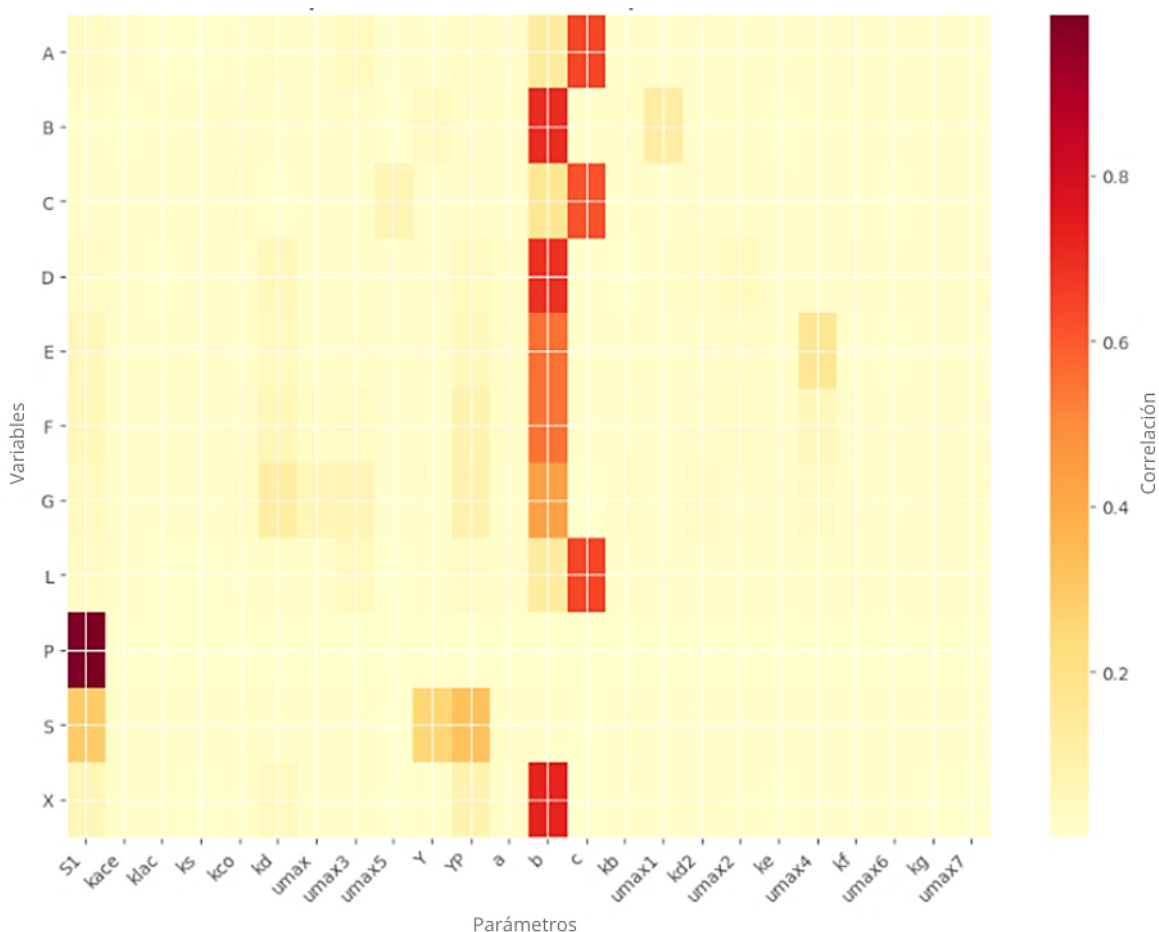


Figura 26. Mapa de calor de sensibilidad normalizada de los parámetros sobre las variables dinámicas del modelo estructurado.

La figura 26 muestra el mapa de calor de sensibilidad normalizada (o importancia relativa) de los parámetros cinéticos sobre las variables dinámicas del modelo ($A, B, C, D, E, F, G, L, P, S, X$). La escala de color representa la magnitud de la sensibilidad donde los valores cercanos a 1 (rojo intenso) indican alta influencia paramétrica y valores cercanos a 0 (amarillo claro) indican baja contribución. Se observa una estructura no uniforme en la matriz de sensibilidad, evidencia de que el sistema no presenta influencia paramétrica homogénea, sino que está dominado por un conjunto reducido de parámetros.

Los parámetros ubicados en la región central del eje horizontal (en las constantes c y b) muestran mayores valores de sensibilidad sobre múltiples variables del sistema, particularmente en la formación de biomasa (X), además de que variables como C, D, E y F muestran dependencia significativa a esas constantes.

Los parámetros con valores cercanos a 0 en la mayoría de variables puede indicar que su variación dentro del rango evaluado no altera significativamente la dinámica, por lo que su calibración experimental no sería de prioridad. Por otro lado, la variable del sulfuro de hidrógeno (P) muestra alta dependencia al parámetro SI y en la variable de biomasa (X) está fuertemente relacionada a uno de los parámetros dominantes (b).

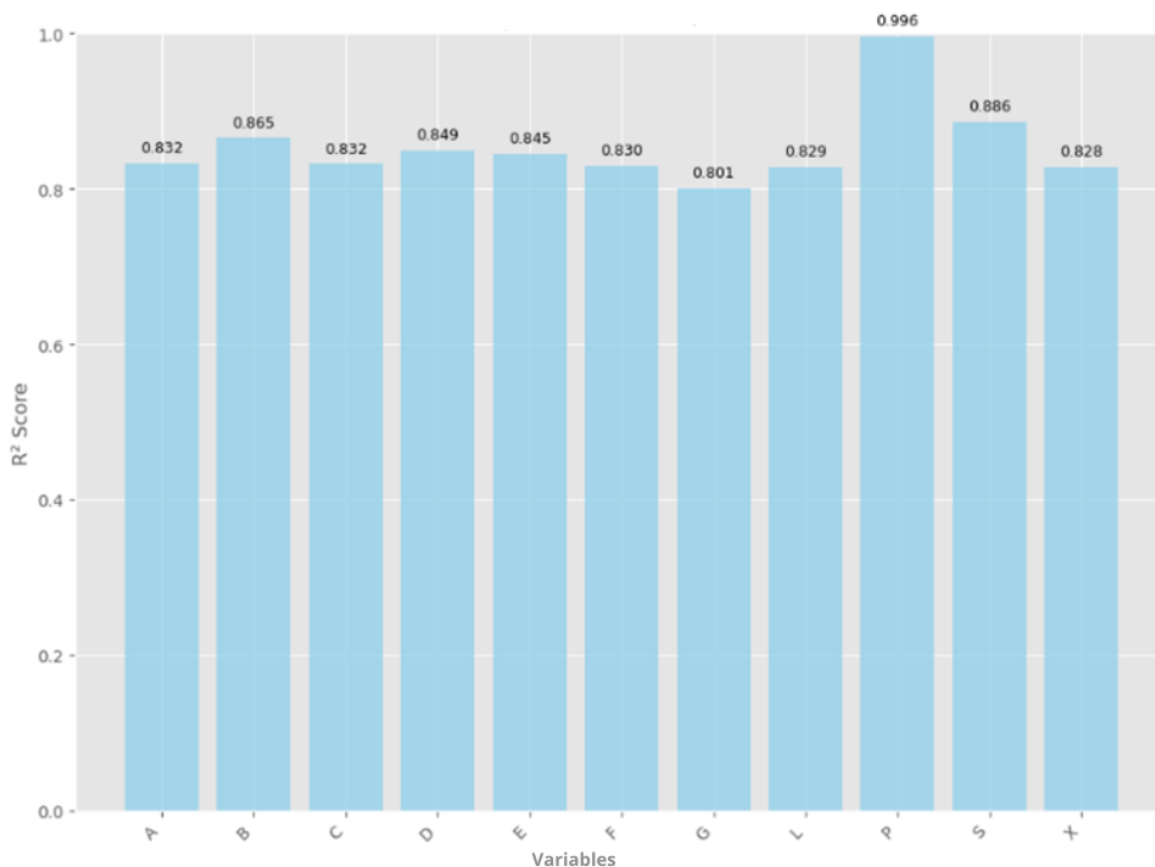


Figura 27. Comparación de los coeficientes de determinación de las variables dinámicas del modelo estructurado.

La figura 27 muestra el coeficiente de determinación (R^2) obtenido para cada variable, con valores que van desde 0.801 a 0.996, lo que indica que el modelo explica entre el 80% y el 99.6% de la variabilidad experimental. La mayoría de las variables presentan valores de R^2 superiores a 0.83, es decir que tiene un desempeño predictivo aceptable. Es importante mencionar que la variable *P* destaca con un valor de R^2 de 0.996, un ajuste casi perfecto. Por otro lado, la variable *G* presenta el valor de R^2 más bajo (0.801), pero sigue dentro del rango aceptable. Estos resultados en conjunto confirman que el modelo reproduce adecuadamente la dinámica experimental de las variables evaluadas, con una mayor precisión en la formación de producto y un desempeño algo menor en algunos intermediarios metabólicos.

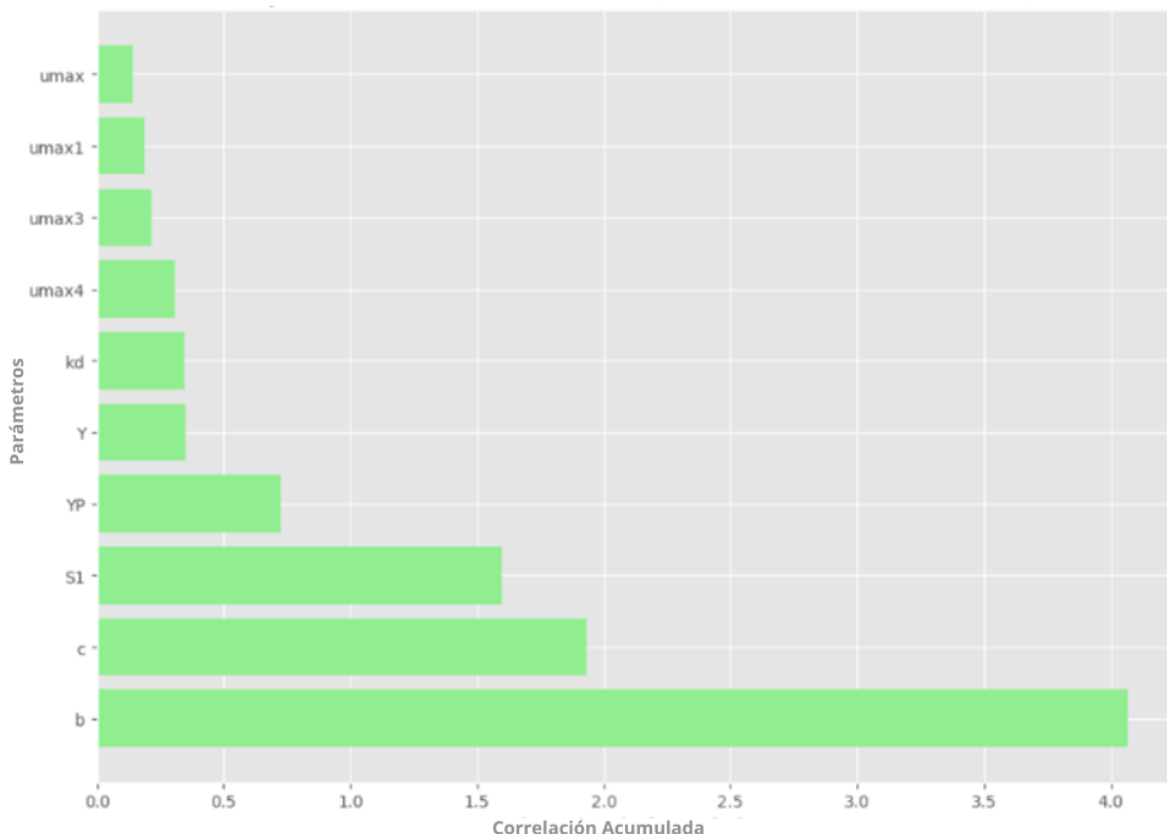


Figura 28. Importancia global de los parámetros más influyentes en el modelo estructurado.

La figura 28 muestra la importancia global de los parámetros, calculada como la suma de sus sensibilidades normalizadas sobre todas las variables de estado del modelo. Se observa claramente una jerarquía en la importancia paramétrica. Aquí el parámetro b presenta la mayor influencia acumulada (ligeramente superior a 4), siendo el principal determinante de la dinámica global del sistema. En segundo lugar de importancia está c y SI , con valores cercanos a 2 y 1.6 respectivamente, lo que indica un efecto significativo sin ser dominante. Por último, para los parámetros YP , Y y k muestran influencia moderada, mientras que para el resto de parámetros presentan contribuciones relativamente similares entre sí y cada vez más bajas. Esto confirma que el modelo posee una estructura paramétrica no uniforme, donde existe un conjunto muy reducido de parámetros dominantes en la dinámica. Desde un punto de vista para calibración e identificabilidad, se deberían priorizar a los parámetros b , c y SI , en ese orden, por tener la mayor sensibilidad global del sistema.

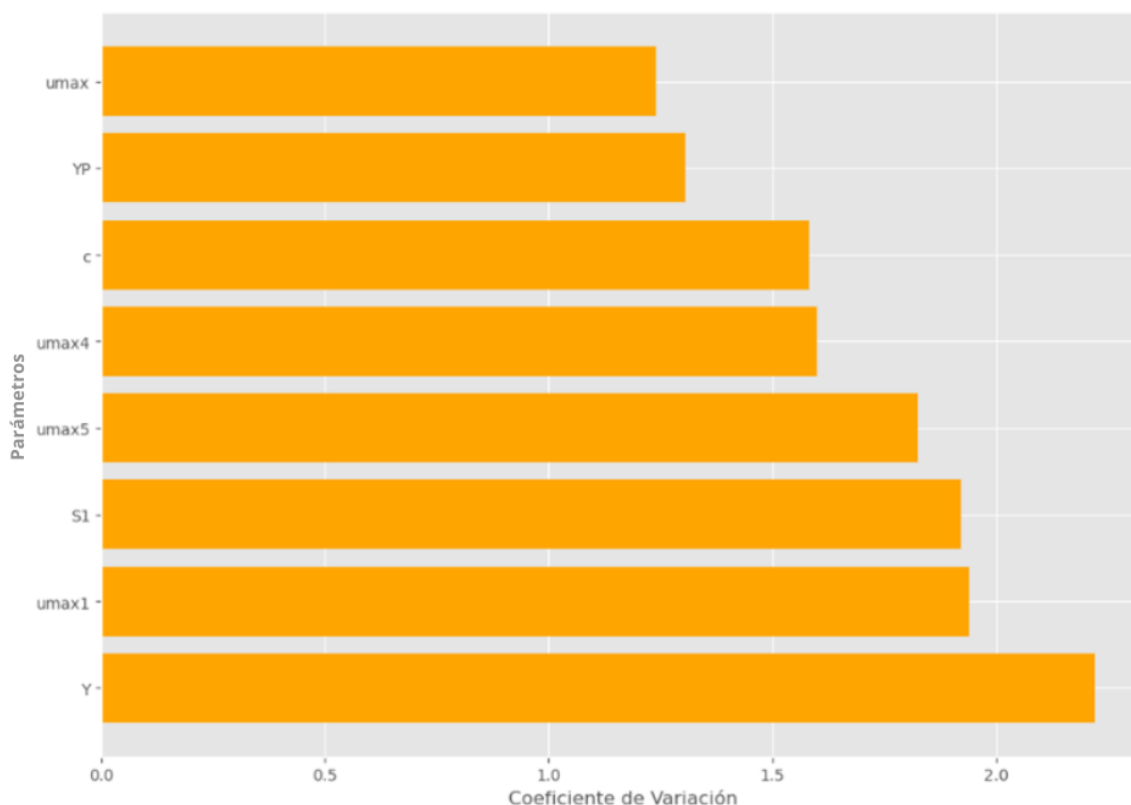


Figura 29. Especificidad paramétrica mediante el coeficiente de variación de las sensibilidades a través de las distintas variables de estado.

Nota: Alta = afecta pocas variables, Baja = afecta muchas variables.

La figura 29 muestra la especificidad paramétrica evaluada mediante el coeficiente de variación de las sensibilidades a través de las distintas variables de estado. En este contexto, los valores altos indican que el parámetro afecta de manera más selectiva a un conjunto reducido de variables (alta especificidad), mientras que los valores bajos muestran un efecto más distribuido sobre las variables (baja especificidad). Se puede observar que el parámetro Y presenta un mayor coeficiente de variación (cerca a 2.2), indicando una alta especificidad a ciertas variables del sistema. De igual manera, μ_{max1} y $S1$ muestran niveles suficientemente altos de especificidad (cerca de 2.0), sugiriendo efectos más focalizados. Por otro lado, parámetros con valores igual o menores a 1.3 indican una influencia más homogénea sobre las variables. Estos resultados dejan muy en claro que el modelo combina parámetros de efecto global con otros de acción más focalizada, esto refuerza la estructura jerárquica observada en el análisis de sensibilidad global.

Tabla 19. Valores de R^2 , MSE, parámetros importantes y el parámetro más influyente con respecto a las variables respuesta del modelo expandido.

Variable respuesta	R^2	MSE (error cuadrático medio)	Principales parámetros importantes	Parámetro más influyente para la variable
A	0.8323	12839.47	c: 0.6380 b: 0.1314 $\mu_{\max 3}$: 0.0413	<i>c</i>
B	0.8654	7253.65	b: 0.7024 $\mu_{\max 1}$: 0.1198 Y: 0.0346	<i>b</i>
C	0.8317	41.24	c: 0.6126 b: 0.1612 $\mu_{\max 5}$: 0.0643	<i>c</i>
D	0.8491	198.96	b: 0.6878 kd: 0.0517 YP: 0.0394	<i>b</i>
E	0.8452	14811.50	b: 0.5500 $\mu_{\max 4}$: 0.1617 S1: 0.0540	<i>b</i>
F	0.8300	4342.92	b: 0.5444 YP: 0.0804 kd: 0.0571	<i>b</i>
G	0.8010	1979.49	b: 0.4310 kd: 0.1176 YP: 0.0882	<i>b</i>
L	0.8285	13129.50	c: 0.6376 b: 0.1320 $\mu_{\max 3}$: 0.0409	<i>c</i>
P	0.9964	23.05	S1: 0.9952 ks: 0.0005 $\mu_{\max 3}$: 0.0003	<i>S1</i>
S	0.8863	19611.66	YP: 0.3223 S1: 0.2937 Y: 0.2545	<i>YP</i>
X	0.8282	50.74	b: 0.7200 YP: 0.0783 S1: 0.0587	<i>b</i>

Fuente: elaboración propia

De acuerdo con la tabla 19, la alta sensibilidad al parámetro *S1* indica que la acumulación del sulfuro es el principal factor limitante, lo que coincide con los reportes de inhibición por H_2S en bacterias sulfato-reductoras (Keller et al., 2014). El lactato actúa como fuente de carbono y energía simultáneamente. La influencia del parámetro *b* (asociado a lactato) indica dependencia crítica del donador de electrones. La sensibilidad de $\mu_{\max}$ y *kd* refleja limitaciones en la

conservación de la energía. Lo que sugiere un posible cuello de botella en la generación de ATP vía fosforilación a nivel de sustrato. En otras palabras, se puede identificar las siguientes tres limitaciones metabólicas de acuerdo a los resultados obtenidos; siendo la primera la limitación por inhibición de producto debido a que el parámetro SI mostró la mayor sensibilidad en general, desde un punto de vista bioquímico sugiere que la acumulación de sulfuro inhibe fuertemente tanto el crecimiento como la actividad metabólica, coincidiendo con las observaciones de Keller (2014) sobre la inhibición por H_2S en *Desulfovibrio*; la segunda limitación es por la disponibilidad de sustrato identificados en la alta sensibilidad de los parámetros b y c asociados al lactato, esto se traduce a que el lactato funciona como donador de electrones y fuente de carbono simultáneamente e implica que la optimización de la concentración de lactato es un punto crítico para el rendimiento; finalmente, la tercera limitación es la energética demostrado en la alta sensibilidad de los parámetros de μ_{max} y kd que sugiere que la eficiencia en la conservación de la energía puede limitar el crecimiento del microorganismo, esto coincide con el mecanismo de ciclado de hidrógeno descrito por Keller et al., (2014). Las estrategias operativas serían la remoción de sulfuro para reducir inhibición por producto, una alimentación controlada de lactato para mantener concentraciones óptimas de crecimiento y la optimización en las condiciones ambientales como lo es el pH y la temperatura para que permitan maximizar μ_{max} y minimizar el estrés celular para reducir el valor de kd . La validación del modelo metabólico para *Oleidesulfovibrio alaskensis* demostró que la transición de un enfoque de 'caja negra' a uno estructurado es fundamental para capturar la complejidad del bioproceso. Mientras que el modelo no estructurado se limita a describir el crecimiento global, el modelo estructurado implementado integra parámetros cinéticos específicos, como una μ_{max} de 4.32 mM/s y una K_m de 0.16 mM para el D-lactato, permitiendo identificar la oxidación hacia piruvato como el paso limitante bajo las variaciones de pH observadas (8.2 a 7.1). Esta robustez se vio fortalecida por el uso de entornos FLOSS y la librería *pyFOOMB* en *Python*, herramientas que no solo garantizan la reproducibilidad de los resultados mediante análisis de sensibilidad global y simulaciones de Monte Carlo, sino que también permitieron una transparencia algorítmica. En consecuencia, este modelo se presenta como una herramienta predictiva de alta fidelidad, capaz de describir la dinámica metabólica interna con una precisión estadística que valida su potencial aplicación en el diseño y optimización de estrategias de biorremediación (López-González et al., 2023; López-González et al., 2024; Mata-Machuca et al., 2025; Peña Caballero et al., 2026; Meneses-Martínez et al., 2026).

Conclusión

Se implementó un modelo cinético dinámico no estructurado para *Oleidesulfovibrio alaskensis* compuesto por un sistema de seis variables de estado y 13 parámetros cinéticos, logrando describir las dinámicas macroscópicas de crecimiento. A partir de la integración de datos multi-ómicos desde plataformas bioinformáticas (KEGG, UniProt, BRENDA, ExPASy y MetaCyc), se realizó la reconstrucción topológica de una propuesta de red metabólica. Esta aproximación permitió trascender la caracterización empírica hacia un marco de modelado mecanístico estructurado, formulado mediante un sistema de Ecuaciones Diferenciales Ordinarias que acopla la cinética enzimática local con los flujos de metabolitos intermediarios. El análisis de grados de libertad confirmó que el sistema inicial se encuentra exactamente determinado ($GL=0$), garantizando una estructura topológica robusta donde las tasas de reacción intracelulares se acoplan unívocamente a los perfiles temporales de concentración. Posteriormente, la transición del modelo hacia una configuración sobre-determinada (índice de redundancia $R = 3$) establece un mecanismo de acotamiento matemático rígido sobre el espacio de estados interno. Este trabajo representa uno de los primeros desarrollos de modelado estructurado in silico para *O. alaskensis* bajo una arquitectura FLOSS reproducible. La integración de análisis de sensibilidad, técnicas de estimación de parámetros y algoritmos de aprendizaje automático permitió mitigar los problemas de identificabilidad cinemática derivados de la heterogeneidad de datos. El modelo resultante exhibe estabilidad numérica e invariancia estequiométrica, consolidándose como una plataforma cuantitativa robusta para el análisis de flujos metabólicos, el escalamiento de bioprocesos y la posterior validación experimental.

Recomendaciones

A partir de esta investigación y debido a las limitaciones identificadas en el mismo, se abren múltiples líneas de investigación futura que pueden fortalecer y ampliar el alcance del modelo desarrollado.

Una de las principales recomendaciones futuras consiste en la validación experimental directa, mediante cultivos controlados de *O. alaskensis* que permitan obtener datos temporales de crecimiento, consumo de lactato y producción de sulfato, pero enfatizando en las concentraciones de metabolitos secundarios de ser posible.

Otro aspecto puede ser el integrar la regulación metabólica a partir de mecanismos de control enzimático, inhibición por productos o posibles efectos de retroalimentación para dar un paso más realista en la dinámica metabólica del microorganismo. Por último, ver si el modelo puede adaptarse a otras bacterias sulfato-reductoras.

Referencias

- Aguilar López, Ricardo, Ruiz Camacho, Beatriz, Neria-González, M. Isabel, Rangel, Eduardo, Santos, Omar and López Pérez, Pablo A.. "State Estimation Based on Nonlinear Observer for Hydrogen Production in a Photocatalytic Anaerobic Bioreactor" *International Journal of Chemical Reactor Engineering*, vol. 15, no. 5, 2017, pp. 20170004.
<https://doi.org/10.1515/ijcre-2017-0004>
- Barbosa, A. C. C., Venceslau, S. S., & Pereira, I. A. C. (2024). DsrMKJOP is the terminal reductase complex in anaerobic sulfate respiration. *Proceedings Of The National Academy Of Sciences*, 121(6), e2313650121. <https://doi.org/10.1073/pnas.2313650121>
- BRENDA. (s.f.) BRENDA: The Comprehensive Enzyme Informtion System. Technische Universitat Braunschweig. <https://www.brenda-enzymes.org/>
- Cheng, C., O'Brien, E. J., McCloskey, D., Utrilla, J., Olson, C., LaCroix, R. A., Sandberg, T. E., Feist, A. M., Palsson, B. O., & King, Z. A. (2019). Laboratory evolution reveals a two-dimensional rate-yield tradeoff in microbial metabolism. *PLoS Computational Biology*, 15(6), e1007066. <https://doi.org/10.1371/journal.pcbi.1007066>
- Coria, S. (s. f.). *Grados De Libertad*. Scribd.
<https://es.scribd.com/document/468472221/GRADOS-DE-LIBERTAD>
- Fazt. (2020, 27 noviembre). *Anaconda - Python Package manager (Conda)* [Video]. YouTube.
<https://www.youtube.com/watch?v=aE7qxfgubS8>

- Frank, S. A. (2010). The trade-off between rate and yield in the design of microbial metabolism. *Journal Of Evolutionary Biology*, 23(3), 609-613. <https://doi.org/10.1111/j.1420-9101.2010.01930.x>
- Garibay Benitez, G. (2017). *Slideshare*. Obtenido de Modelos conéticos estructurados: <https://es.slideshare.net/slideshow/modelos-cinticos-estructurados/76038117>
- Gouveia, C. R. (7 de Agosto de 2020). *Como usar Jupyter notebook, que es jupyter lab [Curso de Python Data Science Español]*. Obtenido de <https://youtu.be/CwbMaSkKDZg?si=Xh49KmWPF1aMIPcZ>
- Gualdron, F. (s.f.). *Modelo De Caja Blanca*. Scribd. <https://es.scribd.com/document/473999349/MODELO-DE-CAJA-BLANCA>
- Hemmerich, J., Tenhaef, N., Wiechert, W., & Noack, S. (2021). pyFOOMB: Python framework for object oriented modeling of bioprocesses. *Engineering In Life Sciences*, 21(3-4), 242-257. <https://doi.org/10.1002/elsc.202000088>
- Heyland, J., Fu, J., & Schmid, A. (2010). Quantification of anaerobic metabolism in sulfate-reducing bacteria using metabolic flux analysis. *Biotechnology and Bioengineering*, 107(4), 693-703.
- Isokpehi, R. D., McInnis, D. S., Destefano, A. M., Johnson, G. S., Walker, A. D., Hall, Y. A., Mapp, B. W., Johnson, M. O., & Simmons, S. S. (2021). Bioinformatics Investigations of Universal Stress Proteins from Mercury-Methylating Desulfovibrionaceae. *Microorganisms*, 9(8), 1780. <https://doi.org/10.3390/microorganisms9081780>
- Jiménez-Perea, M. G. J. M. G., Aguilar-López, L. M. A. L. M., Ayala-Camerena, D. A. D., Rodríguez, S. G., Esquivel, F. M., Mendoza, L. P., & Peña-Caballero, V. (2023, 9 agosto). *A simple elemental balance for the anaerobic sulfate removal by Desulfovibrio*

alaskensis 6SR.

<https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/4140>

Keller, K. L., Rapp-Giles, B. J., Semkiw, E. S., Porat, I., Brown, S. D., & Wall, J. D. (2013).

New Model for Electron Flow for Sulfate Reduction in *Desulfovibrio alaskensis* G20.

Applied And Environmental Microbiology, 80(3), 855-868.

<https://doi.org/10.1128/aem.02963-13>

Keller, K. L., Wall, J. D., & Chhabra, S. (2014). New model for electron flow for sulfate

reduction in *Desulfovibrio alaskensis* G20. *Applied and Environmental Microbiology*, 80(2), 760-768.

Keskin, Y. (2024). *Biocorrosion of copper by Oleidesulfovibrio alaskensis* G20 biofilms in static and dynamic environments. Obtenido de A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Chemical Engineering. MONTANA STATE UNIVERSITY. Bozeman, Montana.:

<https://scholarworks.montana.edu/items/88e5d651-0ff2-4d55-861b-812ba247e44a>

Lipson, D. A. (2015). The complex relationship between microbial growth rate and yield and its implications for ecosystem processes. *Frontiers In Microbiology*, 6, 615.

<https://doi.org/10.3389/fmicb.2015.00615>

Liu, N., Ostertag-Henning, C., Fernø, M. A., & Dopffel, N. (2025). Growth on Hydrogen by the

Sulfate-Reducing *Oleidesulfovibrio alaskensis* Induces Biofilm Dispersion and

Detachment—Implications for Underground Hydrogen Storage. *Environmental Science & Technology*, 59(14), 7095-7105. <https://doi.org/10.1021/acs.est.4c13893>

López-González, I., Lucho-Constantino, C., Beltrán-Hernández, R. & López-Pérez, P. (2025).

Simulation-based assessment of anaerobic digestion scenarios for sustainable valorization

of *Sargassum* spp. biomass. Chemical Product and Process Modeling.

<https://doi.org/10.1515/cppm-2025-0214>

López-González, I.E., López-Pérez, P.A. and Hernández-Melchor, D.J. (2024). Coupling of Process Intensification to Life Cycle Assessment for Eco-Design of Biohydrogen Factory by Microalgae. In *Towards Green Hydrogen Generation* (eds M. Sankir and N.D. Sankir). <https://doi.org/10.1002/9781394234110.ch6>

López-Pérez, P. A. (2025). State estimation for novel kinetic model oscillatory behavior with a time delay for bioethanol production. *Instrumentation Science & Technology*, 53(6), 735–759. <https://doi.org/10.1080/10739149.2025.2470134>

López Pérez P.A., O.J. Santos Sánchez, L. Rodríguez Guerrero and O. Patricio. Advanced Control Methods for Industrial Processes. In *Advanced Control Methods for Industrial Processes* (eds P.A. López Pérez, O.J. Santos Sánchez, L. Rodríguez Guerrero and O. Patricio). Wiley, (2025). <https://doi.org/10.1002/9783527844197.fmatter>

López Pérez (2013). *Diseño de un bioproceso de alta eficiencia para la remoción de Cadmio vía Desulfovibrio alaskensis 6SR: modelado, simulación, estimación y control*. Obtenido de [Tesis de Doctorado, Centro de Investigación y de estudios avanzados del Instituto Politecnico Nacional]

López Pérez, P.A., Neria González, M.I. & Aguilar López, R. Cadmium concentration stabilization in a continuous sulfate reducing bioreactor via sulfide concentration control. *Chem. Pap.* 67, 326–335 (2013). <https://doi.org/10.2478/s11696-012-0274-8>

López Pérez, P.A., Aguilar López, R. & Neria González, M.I. Cadmium removal at high concentration in aqueous medium: mediated by *Desulfovibrio alaskensis* . *Int. J. Environ. Sci. Technol.* 12, 1975–1986 (2015). <https://doi.org/10.1007/s13762-014-0601-4>

- López Pérez P.A., M.I. Neria González, M.R. Pérez García, R. Aguilar López, Concentrations monitoring via software sensor for bioreactors under model parametric uncertainty: Application to cadmium removal in an anaerobic process, *Alexandria Engineering Journal*, Volume 55, Issue 2, 2016, 1893-1902, <https://doi.org/10.1016/j.aej.2016.03.013>
- Mata-Machuca, J. L., López-Pérez, P. A., & Aguilar-López, R. (2025). Novel Nonlinear Control in a Chaotic Continuous Flow Enzymatic–Fermentative Bioreactor. *Fermentation*, 11(10), 601. <https://doi.org/10.3390/fermentation11100601>
- Manhart, M., Adkar, B. V., & Shakhnovich, E. I. (2018). Trade-offs between microbial growth phases lead to frequency-dependent and non-transitive selection. *Proceedings Of The Royal Society B Biological Sciences*, 285(1872), 20172459. <https://doi.org/10.1098/rspb.2017.2459>
- Márquez, D. E. Á. (2022). Principales microorganismos sulfato reductores (MSR) de reactores anaerobios alimentados con efluentes ácidos, una revisión bibliográfica. *Revista Colombiana de Biotecnología*, 24(1), 62-76. <https://doi.org/10.15446/rev.colomb.biote.v24n1.100963>
- Meneses-Martínez, P., González-Viveros, I., Ordaz, P., Aguilar-López, R., López-Pérez, P. A., & Mata-Machuca, J. L. (2026). Dedicated Observers for Sensors Fault Detection and Diagnosis in Real Time for Bioreactors. *Sensors*, 26(4), 1095. <https://doi.org/10.3390/s26041095>
- Mori, M., Marinari, E., & De Martino, A. (2017). *A yield-cost tradeoff governs Escherichia coli's decision between fermentation and respiration in carbon-limited growth*. <https://www.semanticscholar.org/paper/A-yield-cost-tradeoff-governs-Escherichia-coli%E2%80%99s-in-Mori->

Marinari/24a765654900f4981311177eec23560e342f9940?sort=is-influential#citing-papers

Neria Gonzalez M. A., A. L. (s.f.). Revisión de los procesos de remoción y recuperación de los metales pesados a través de *Desulfovibrio alaskensis* 6SR, bacteria sulfato reductora. *Tecnocultura*, 17 - 31.

Neria-González, M. I., López-Pérez, P. A., Aguilar-López, R., Partial Control of a Continuous Bioreactor: Application to an Anaerobic System for Heavy Metal Removal, *Journal of Engineering*, 2016, 4358282, 7 pages, 2016. <https://doi.org/10.1155/2016/4358282>

Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (s.f.). *Scikit-learn: Machine Learning in Python*. Obtenido de *Journal of Machine Learning Research* : https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?source=post_page

Peña-Caballero Vicente, Ricardo Aguilar-López, Pablo A. López-Pérez, M. Isabel Neria-González, Reduction of Cr(VI) utilizing biogenic sulfide: an experimental and mathematical modeling approach, *Desalination and Water Treatment*, Volume 57, Issue 28, 2016, Pages 13056-13065, <https://doi.org/10.1080/19443994.2015.1055811>.

Peña Caballero, V., Rodríguez-Mata, A. E., López-Pérez, P. A., Hernández-Melchor, D. J., & González-Huitrón, V. A. (2026). A Robust State Estimation Framework Employing a Nonlinear PI2 Observer for Photobioreactor Monitoring. *AppliedMath*, 6(3), 44. <https://doi.org/10.3390/appliedmath6030044>

Pereira, I. A. C., Ramos, A. R., Grein, F., Marques, M. C., da Silva, S. M., & Venceslau, S. S. (2011). A unifying concept for the metabolic energy conservation of sulfate-reducing prokaryotes. *Frontiers in Microbiology*, 2, 24.

- Pérez, P.A.L., Martínez, P.M., Castañeda, E.V. and López, R.A. (2024). In Situ Monitoring for Biohydrogen Production Using a Low-Cost Sensor . In Towards Green Hydrogen Generation (eds M. Sankir and N.D. Sankir). <https://doi.org/10.1002/9781394234110.ch7>
- Petrovsky, D. V., Malsagova, K. A., Rudnev, V. R., Kulikova, L. I., Pustovoyt, V. I., Balakin, E. I., Yurku, K. A., & Kaysheva, A. L. (2023). Bioinformatics Methods for constructing metabolic networks. *Processes*, 11(12), 3430. <https://doi.org/10.3390/pr11123430>
- Rafa González Gouveia. (2020, 7 agosto). *Cómo Usar Jupyter Notebook qué es Jupyterlab [Curso Python Data Science Español]* [Video]. YouTube. <https://www.youtube.com/watch?v=CwbMaSkKDZg>
- Raya, D., Peta, V., Gadhamshetty, V., Gnimpieba, E. Z., & Dhiman, S. S. (2026). Genome announcement of sulfate-reducing *Oleidesulfovibrio alaskensis* G20 with novel electron transfer gene annotations. *Microbiology Resource Announcements*, 15(4), e0094225. <https://doi.org/10.1128/mra.00942-25>
- Saxena, P., Samanta, D., Thakur, P., Goh, K. M., Subramaniam, M., Peyton, B. M., Fields, M., & Sani, R. K. (2025). pH-dependent genotypic and phenotypic variability in *Oleidesulfovibrio alaskensis* G20. *Applied And Environmental Microbiology*, 91(4), e0256524. <https://doi.org/10.1128/aem.02565-24>
- Singh, R. N., & Sani, R. K. (2024). Genome-Wide Computational Prediction and Analysis of Noncoding RNAs in *Oleidesulfovibrio alaskensis* G20. *Microorganisms*, 12(5), 960. <https://doi.org/10.3390/microorganisms12050960>
- Sokolova, D. S., Semenova, E. M., Grouzdev, D. S., Bidzhieva, S. K., Babich, T. L., Loiko, N. G., Ershov, A. P., Kadnikov, V. V., Beletsky, A. V., Mardanov, A. V., Zhaparov, N. S., & Nazina, T. N. (2021). Sulfidogenic Microbial Communities of the Uzen High-

Temperature Oil Field in Kazakhstan. *Microorganisms*, 9(9), 1818.

<https://doi.org/10.3390/microorganisms9091818>

Tang, Y. J., Meadows, A. L., Kirby, J., & Keasling, J. D. (2007). Anaerobic central metabolic pathways in *Desulfovibrio vulgaris* Hildenborough detected by assisted metabolite profiling. *Journal of Bacteriology*, 189(3), 894-903

UniProt. (s. f.). UniProt. <https://www.uniprot.org/uniprotkb/Q30WB5/entry>

Vita, N., Valette, O., Brasseur, G., Lignon, S., Denis, Y., Ansaldi, M., Dolla, A., & Pieulle, L. (2015). The primary pathway for lactate oxidation in *Desulfovibrio vulgaris*. *Frontiers In Microbiology*, 6, 606. <https://doi.org/10.3389/fmicb.2015.00606>

Apéndices

MODELO NO ESTRUCTURADO

Apéndice A

Apéndice B

MODELO ESTRUCTURADO

Apéndice C

Apéndice D

Apéndice E

5.1 Apéndice A

Código 1 de modelado no estructurado

Modelo para datos de lactato, acetato, sulfato, sulfuro, biomasa y carbono mediante Jupyter Notebook con el paquete pyFOOMB para la obtención de la figura 13, del apartado de resultados.

```
[3]: # imports from the pyfoomb package
import pyfoomb
print(f'Current package version of {pyfoomb.__name__}: {pyfoomb.__version__}')

from pyfoomb import BioprocessModel
from pyfoomb import ObservationFunction
from pyfoomb import Caretaker
from pyfoomb import Visualization

[23]: # Define la clase de modelo
class SequentialKinetic(BioprocessModel):

    def rhs(self, t, y):
        # Descomprime el vector de estado. Los estados son ordenados alfabeticamente. #pyx11 para extraer datos
        A, C, L, P, S, X = y

        # Descomprime los parámetros del modelo. Ejemplo:
        S1 = self.model_parameters['S1']
        kace = self.model_parameters['kace']
        klac = self.model_parameters['klac']
        ks = self.model_parameters['ks']
        kco = self.model_parameters['kco']
        kd = self.model_parameters['kd']
        #velocidades
        umax = self.model_parameters['umax']
        umax4 = self.model_parameters['umax4']
        umax5 = self.model_parameters['umax5']
        #rendimientos
        Y = self.model_parameters['Y']
        YP = self.model_parameters['YP']
        #exponentes
        a = self.model_parameters['a']
        b = self.model_parameters['b']

        c = self.model_parameters['c']

        # Define las derivadas.
        dSdt = -(Y)*(umax)*((1-(P/S1)**a)*(S/(ks+S)))*X*(L**b)
        dPdt = (umax*(YP))*(((1-(P/S1)**a)*(S/(ks+S)))*X*(L**b))
        dXdt = (umax*((1-(P/S1)**a)*(S/(ks+S)))*X)-(kd*X*(L**b))
        dLdt = (-umax4)*((L**c)/(klac+L**c))*X*(kace/(A+kace))
        dAdt = (umax4)*((L**c)/(klac+L**c))*X*(kace/(A+kace))
        dCdt = (umax5)*((L**c)/(klac+L**c))*X*(kco/(kco+C))

        # Regresa las derivadas como lista (or numpy array)
        # El orden corresponde al vector de estado.
        return [dAdt, dCdt, dLdt, dPdt, dSdt, dXdt]

        # Define un diccionario para los valores iniciales.
        # La clave corresponde al estado del modelo, extendidos por un 0 (cero).
```

```

initial_values = {
    'A0':0.1,
    'C0':0.1,
    'L0':4640,
    'P0':29,
    'S0':5655,
    'X0':5,
}

# Define un diccionario para los parametros del modelo.
# La clave se relaciona con los nombres de las variables utilizadas en la clase del modelo. Ejemplo:
    # 'k_S' : 0.01,
    # 'mu_max' : 0.4,
    # 'Y_PX' : 0.4,
    # 'Y_XS' : 0.5,
model_parameters = {
    #constantes cinéticas
    'S1':680,
    'kace':1500,
    'klac':95000,
    'ks':0.0009048,
    'kco':900,
    'kd':0.04,

    #velocidades
    'umax':0.13,
    'umax4':0.6,
    'umax5':0.12,
    #rendimientos
    'Y':7.5,
    'YP':1,
    #exponentes
    'a':1.9,
    'b':0.001,
    'c':1.5,
}

# Instanciar un Caretaker, que administra el modelo y su parametrización.
caretaker = Caretaker(
    bioprocess_model_class=SequentialKinetic,
    model_parameters=model_parameters,
    initial_values=initial_values,
)

# Ejecuta una simulación hacia adelante.
simulation = caretaker.simulate(t=216)

# Instanciar un Caretaker, que administra el modelo y su parametrización.
caretaker = Caretaker(
    bioprocess_model_class=SequentialKinetic,
    model_parameters=model_parameters,
    initial_values=initial_values,
)

# Ejecuta una simulación hacia adelante.
simulation = caretaker.simulate(t=216)

# Visualizar las trayectorias resultantes.
_ = Visualization.show_kinetic_data(simulation)

```

def rhs(self, t, y):

representa matemáticamente: $dt/dy=f(t,y,\theta)$

donde:

$y=(A,B,C,D,E,F,G,L,P,S,X) \rightarrow$ vector de estados

$t \rightarrow$ tiempo

$\theta \rightarrow$ parámetros del modelo

$f \rightarrow$ la función que tú defines en rhs

Entonces: $rhs = f(t, y, \theta)$ Es una función vectorial no lineal. “rhs” en este código es un sistema de 11 ecuaciones diferenciales acopladas y no lineales. Combina varias funciones matemáticas que son la cinética tipo Monod, inhibición no lineal, dependencias de potencia y el acoplamiento entre variables.

Código 2: sensibilidad de parámetros

Matemáticamente, la sensibilidad de la variable de estados $x_i(t)$ respecto al parámetro p_j es la derivada parcial $\frac{dx_i(t)}{dp_j}$. Como no se dispone de una expresión analítica, pyfoomb la aproxima numéricamente mediante diferencias finitas centrales:

$$\frac{dx_i(t)}{dp_j} \approx \frac{x_i(t, p_j + \Delta p_j) - x_i(t, p_j - \Delta p_j)}{2\Delta p_j}$$

Donde el tamaño de la perturbación Δp_j se puede definir de dos formas: 1) Relativa: $\Delta p_j = rel_h * \Delta p_j$ (por defecto `rel_h = 1e-3`), que puede ser modificable; y 2) Absoluta: $\Delta p_j = abs_h$ (valor fijo).

A partir de ahí, se calculan las sensibilidades para todas las combinaciones de estados contra parámetros. En la visualización se organizan en columnas de cuatro columnas para facilitar su inspección. Este tipo de grafico es denso, pero permite una visión global de la influencia de cada parámetro. Este análisis es fundamental en ingeniería de bioprocesos para mejorar la identificación de los modelos y optimizar la obtención de datos experimentales.

```
# importar desde el paquete de pyfoomb
import pyfoomb
print(f'Current package version of {pyfoomb.__name__}: {pyfoomb.__version__}')
```

```
from pyfoomb import BioprocessModel
from pyfoomb import ObservationFunction
from pyfoomb import Caretaker
from pyfoomb import Visualization
from pyfoomb import Measurement
```

```
Current package version of pyfoomb: 2.17.7
```

```
from matplotlib import pyplot
import numpy
import pandas
```

```
# Set seed of random number generator for reproducibility
numpy.random.seed(123)
```

```
# Define la clase de modelo
```

```
class ComplexKinetic(BioprocessModel):
```

```
    def rhs(self, t, y):
```

```
        # Descomprime el vector de estado. Los estados son ordenados alfabeticamente. #pyx11 para extraer datos
        A, C, L, P, S, X = y
```

```
        # Descomprime los parámetros del modelo. Ejemplo:
```

```
        S1 = self.model_parameters['S1']
        kace = self.model_parameters['kace']
        klac = self.model_parameters['klac']
        ks = self.model_parameters['ks']
        kco = self.model_parameters['kco']
        kd = self.model_parameters['kd']
```

```
        #velocidades
```

```
        umax = self.model_parameters['umax']
        umax4 = self.model_parameters['umax4']
        umax5 = self.model_parameters['umax5']
```

```
        #rendimientos
```

```
        Y = self.model_parameters['Y']
        YP = self.model_parameters['YP']
```

```
        #exponentes
```

```
        #a = self.model_parameters['a']
        #b = self.model_parameters['b']
        #c = self.model_parameters['c']
```

```
        # Define las derivadas.
```

```
        dSdt = -(Y)*(umax)*((1-(P/S1)**1.9)*(S/(ks+S)))*X*(L**0.001)
        dPdt = (umax*(YP))*(((1-(P/S1)**1.9)*(S/(ks+S)))*X*(L**0.001)
        dXdt = (umax*(1-(P/S1)**1.9)*(S/(ks+S)))*X)-(kd*X*(L**0.001))
        dLdt = (-umax4)*((L**1.5)/(klac+L**1.5))*X*(kace/(A+kace))
        dAdt = (umax4)*((L**1.5)/(klac+L**1.5))*X*(kace/(A+kace))
        dCdt = (umax5)*((L**1.5)/(klac+L**1.5))*X*(kco/(kco+C))
```

```
        # Regresa las derivadas como lista (or numpy array)
```

```
        # El orden corresponde al vector de estado.
```

```
        return [dAdt, dCdt, dLdt, dPdt, dSdt, dXdt]
```

```
# Define un diccionario para los valores iniciales.
```

```
# La clave corresponde al estado del modelo, extendidos por un 0 (cero).
```

```
#
```

```
initial_values = {
    'A0':0.1,
    'C0':0.1,
    'L0':4640,
    'P0':29,
    'S0':5655,
    'X0':5,
}
```

```
# Define un diccionario para los parametros del modelo.
# La clave se relaciona con los nombres de las variables utilizadas en la clase del modelo. Ejemplo:
```

```
model_parameters = {
    #constantes cinéticas
    'S1':680,
    'kace':1500,
    'klac':95000,
    'ks':0.0009048,
    'kco':900,
    'kd':0.04,
    #velocidades
    'umax':0.13,
    'umax4':0.6,
    'umax5':0.12,
    #rendimientos
    'Y':7.5,
    'YP':1,
    #exponentes
    #'a':1.9,
    #'b':0.001,
    #'c':1.5,
}
```

```
# Instanciar un Caretaker, que administra el modelo y su parametrización.
```

```
caretaker = Caretaker(
    bioprocess_model_class=ComplexKinetic,
    model_parameters=model_parameters,
    initial_values=initial_values,
)
```

```
# Correnos una simulación hacia adelante.
```

```
simulation = caretaker.simulate(t=216)
```

```
# Mostrar los resultados de las trayectorias.
```

```
_ = Visualization.show_kinetic_data(simulation)
```

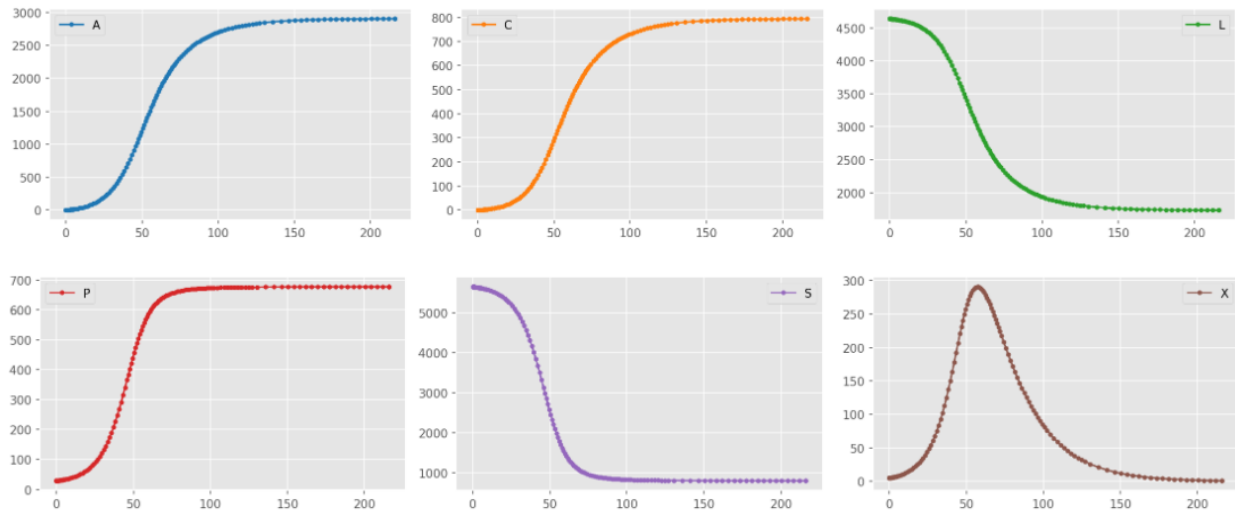
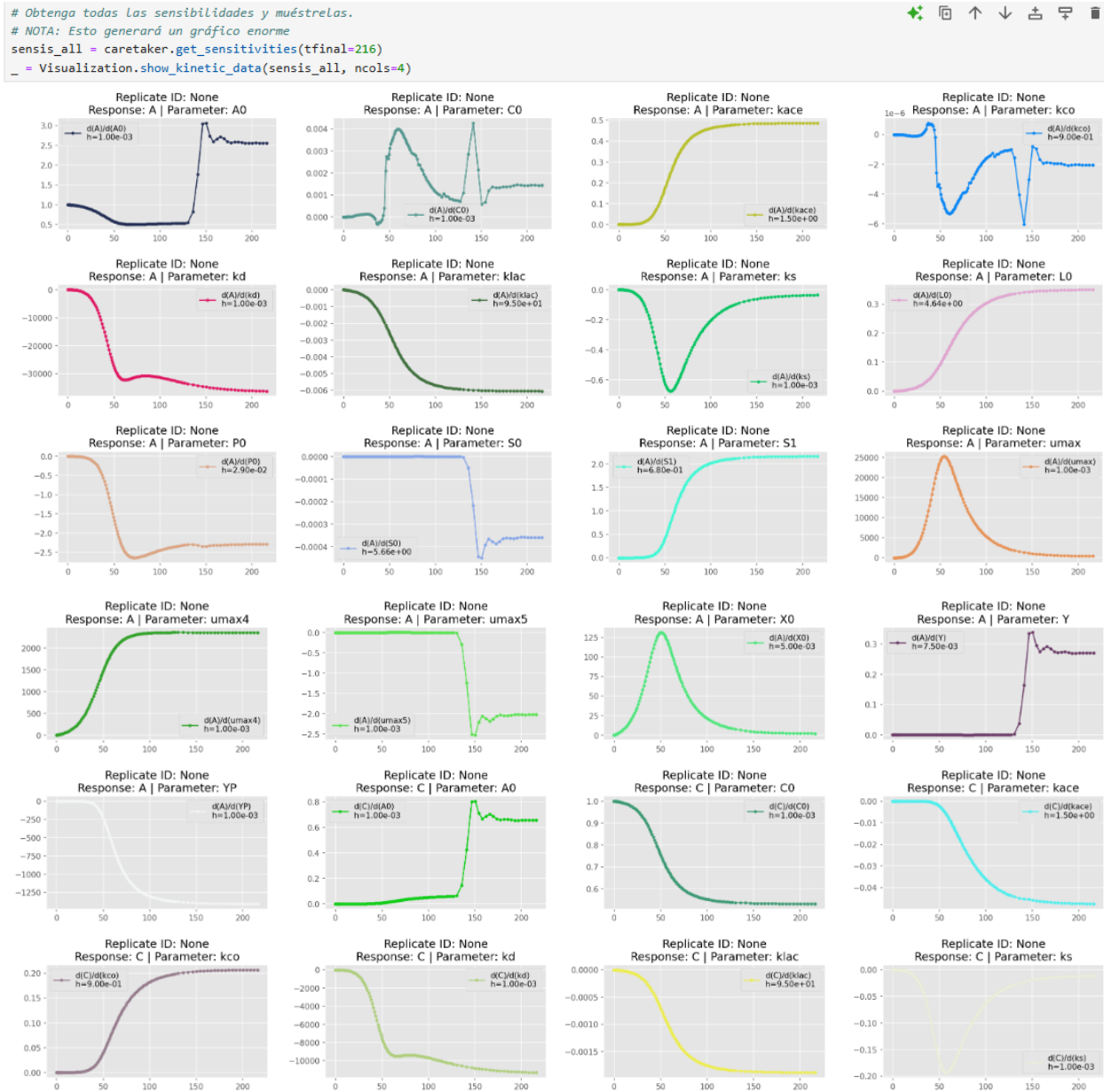
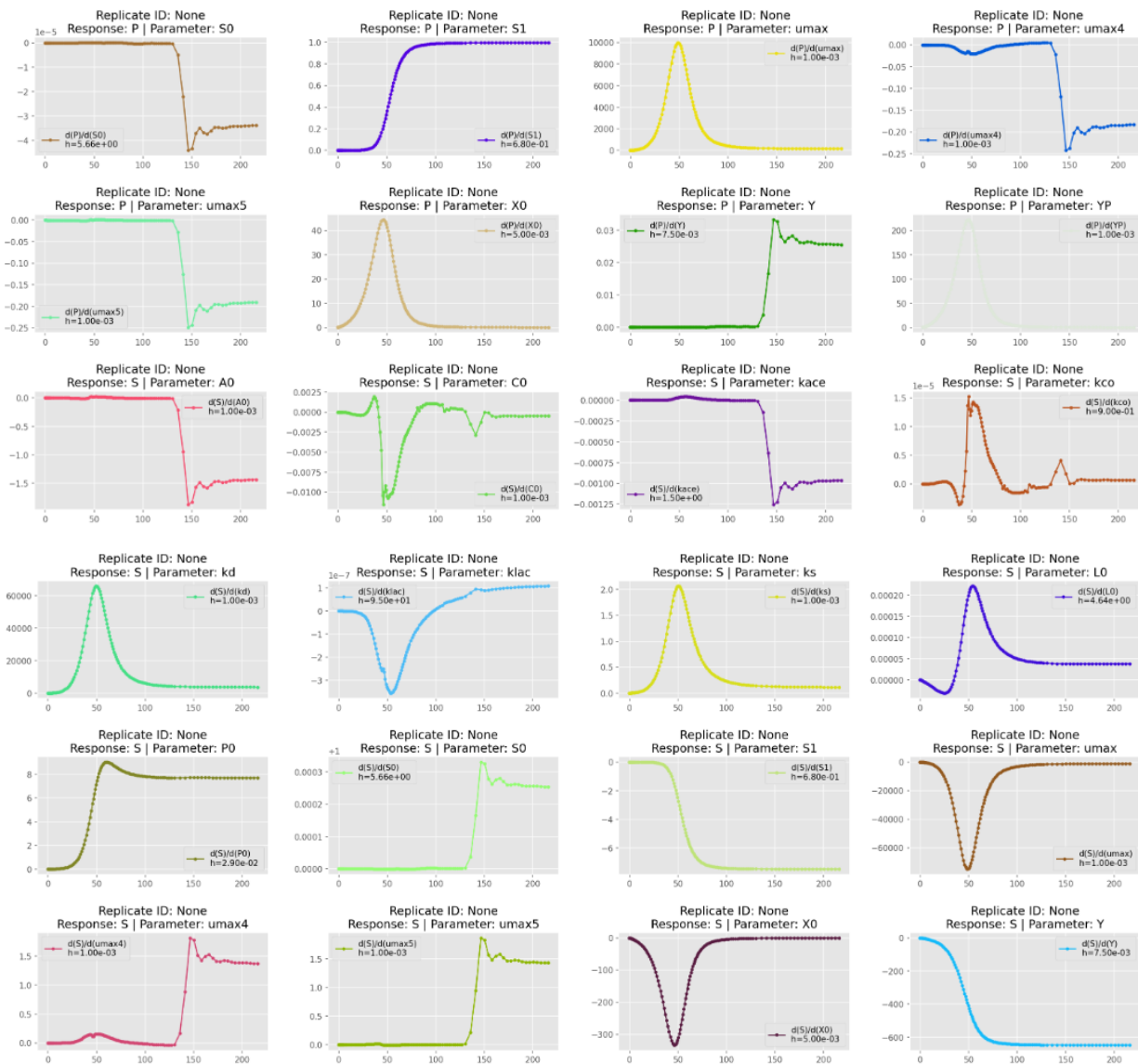
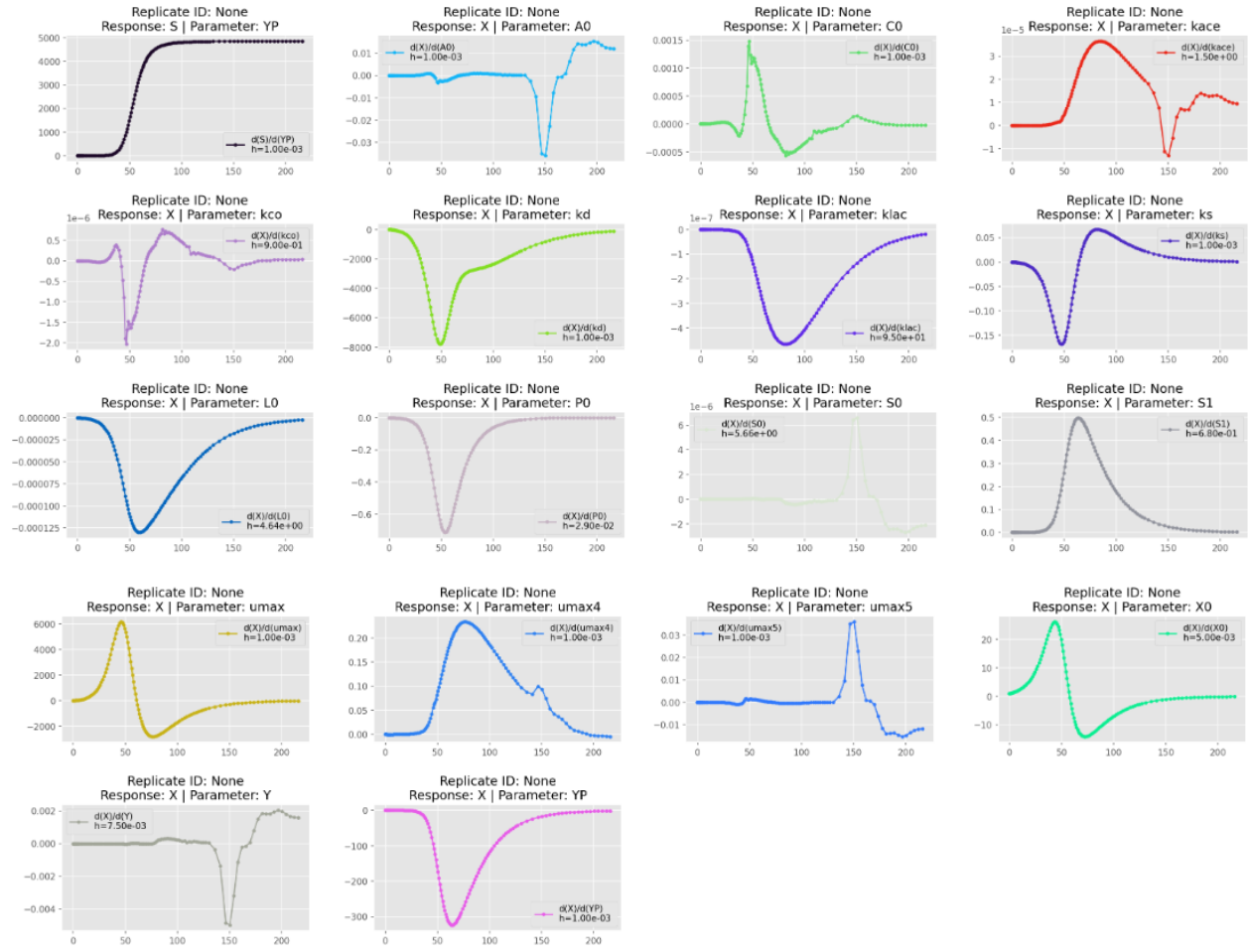


Figura A 30. Resultados del código para el análisis de sensibilidad del modelo no estructurado.







Código 3 en Rstudio

Código para la realizacion del mapa de calor a partir de la tabla de sensibilidades con valores numericos para la obtencion de la figura 14 de la seccion de resultados..

```
1 install.packages("pheatmap")
2 library(pheatmap)
3 library(readxl)
4
5 datos_rcorta <- as.data.frame(Sensibilidad_ruta_meta_corta)
6 rownames(datos_rcorta) <- datos_rcorta[,1]
7 datos_rcorta <- datos_rcorta[,-1]
8 str(datos_rcorta)
9
10 pheatmap(datos_rcorta,
11          main = "Sensibilidad entre variables",
12          display_numbers = TRUE,
13          number_color = "black",
14          fontsize_number = 10,
15          breaks = c(0,1,2,3,4,5),
16          legend_breaks = 1:5,
17          border_color = "gray",
18          cluster_rows = TRUE,
19          cluster_cols = TRUE,
20          color = colorRampPalette(c("yellow", "orange", "red"))(5))
21
```

5.2 Apéndice B

Código 4: Análisis de Monte Carlo.

```
[2]: # importamos desde el paquete de pyfoomb
import pyfoomb
print(f'Current package version of {pyfoomb.__name__}: {pyfoomb.__version__}')

from pyfoomb import BioprocessModel
from pyfoomb import ObservationFunction
from pyfoomb import Caretaker
from pyfoomb import Visualization
from pyfoomb import Measurement

Current package version of pyfoomb: 2.17.7

[3]: from matplotlib import pyplot
import numpy
import pandas

# Establecer la semilla del generador de números aleatorios para la reproducibilidad
numpy.random.seed(123)

[4]: # Define la clase de modelo
class SequentialKinetic(BioprocessModel):

    def rhs(self, t, y):
        # Descomprime el vector de estado. Los estados son ordenados alfabéticamente. #pyx11 para extraer datos
        A, C, L, P, S, X = y

        # Descomprime los parámetros del modelo. Ejemplo:
        S1 = self.model_parameters['S1']
        kace = self.model_parameters['kace']
        klac = self.model_parameters['klac']
        ks = self.model_parameters['ks']
        kco = self.model_parameters['kco']
        kd = self.model_parameters['kd']
        #velocidades
        umax = self.model_parameters['umax']
        umax4 = self.model_parameters['umax4']

        umax5 = self.model_parameters['umax5']
        #rendimientos
        Y = self.model_parameters['Y']
        YP = self.model_parameters['YP']
        #exponentes
        a = self.model_parameters['a']
        b = self.model_parameters['b']
        c = self.model_parameters['c']

        # Define las derivadas.
        dSdt = -(Y)*(umax)*((1-(P/S1)**a)*(S/(ks+S)))*X*(L**b)
        dPdt = (umax*(YP))*(((1-(P/S1)**a)*(S/(ks+S)))*X*(L**b))
        dXdt = (umax*((1-(P/S1)**a)*(S/(ks+S)))*X)-(kd*X*(L**b))
        dLdt = (-umax4)*((L**c)/(klac+L**c))*X*(kace/(A+kace))
        dAdt = (umax4)*((L**c)/(klac+L**c))*X*(kace/(A+kace))
        dCdt = (umax5)*((L**c)/(klac+L**c))*X*(kco/(kco+C))

        # Regresa las derivadas como lista (or numpy array)
        # El orden corresponde al vector de estado.
        return [dAdt, dCdt, dLdt, dPdt, dSdt, dXdt]

# Define un diccionario para los valores iniciales.
# La clave corresponde al estado del modelo, extendidos por un 0 (cero).
```

```

initial_values = {
    'A0':0.1,
    'C0':0.1,
    'L0':4640,
    'P0':29,
    'S0':5655,
    'X0':5,
}

# Define un diccionario para los parametros del modelo.
# La clave se relaciona con los nombres de las variables utilizadas en la clase del modelo. Ejemplo:
model_parameters = {
    #constantes cinéticas
    'S1':680,
    'kace':1500,
    'klac':95000,
    'ks':0.0009048,
    'kco':900,
    'kd':0.04,
    #velocidades
    'umax':0.13,
    'umax4':0.6,
    'umax5':0.12,

    #rendimientos
    'Y':7.5,
    'Yp':1,
    #exponentes
    'a':1.9,
    'b':0.001,
    'c':1.5,
}

```

[5]: # Define un método auxiliar que crea un conjunto de datos ruidoso, basado en una simulación hacia adelante.

```

def noisy_samples(simulations, samples=5, rel_err=0.02, abs_err=0.02, repetitions=3, with_errors=True):
    artificial_data = []
    for simulation in simulations:
        _name = simulation.name
        _replicate_id = simulation.replicate_id
        _timepoints = simulation.timepoints
        _values = simulation.values
        _rnd_values = [
            numpy.random.normal(loc=_values, scale=abs(_values*rel_err+abs_err))
            for _ in range(repetitions)
        ]
        _values = numpy.mean(_rnd_values, axis=0)
        _errors = numpy.std(_rnd_values, ddof=1, axis=0)
        selector = numpy.sort(
            numpy.random.choice(range(len(_timepoints)), size=samples, replace=False),
        )
        if with_errors:
            artificial_data.append(
                Measurement(name=_name, timepoints=_timepoints[selector], values=_values[selector], errors=_errors[selector], replicate_id=_replicate_id),
            )
        else:
            artificial_data.append(
                Measurement(name=_name, timepoints=_timepoints[selector], values=_values[selector], replicate_id=_replicate_id),
            )
    return artificial_data

```

[10]: # Crea una instancia de Caretaker, que administre el modelo y su parametrización.

```

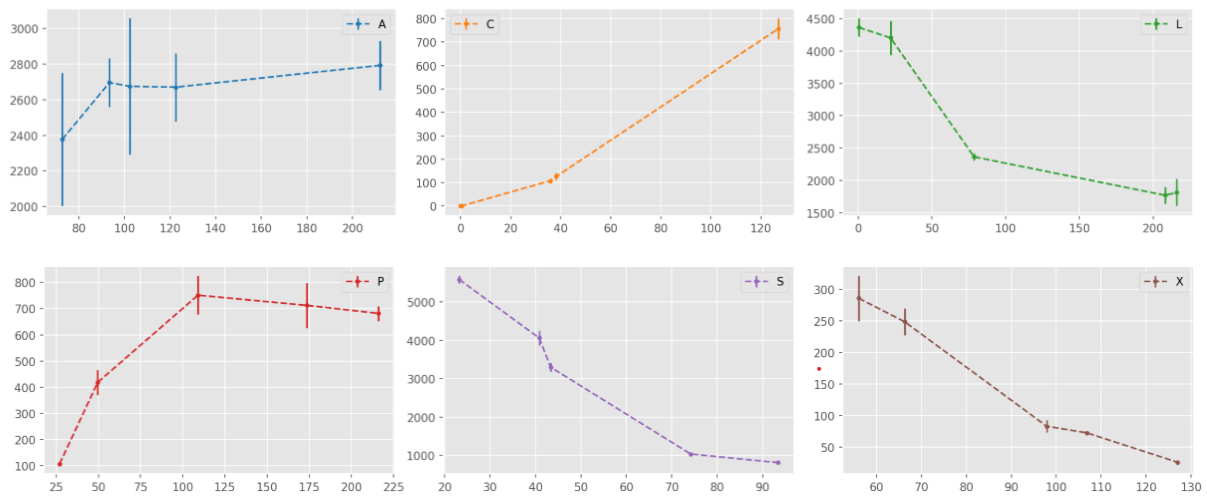
caretaker = Caretaker(
    bioprocess_model_class=SequentialKinetic,
    model_parameters=model_parameters,
    initial_values=initial_values,
)

# Corre una simulación hacia adelante.
simulation = caretaker.simulate(t=216)

```

```
[12]: # Crea los datos ruidosos
data = noisy_samples(simulation, rel_err=0.1)
```

```
[14]: # Visualizar los datos ruidosos
_ = Visualization.show_kinetic_data(data)
```



```
[15]: unknowns = ['S1', 'kace', 'klac', 'ks', 'kco', 'kd', 'umax', 'umax4', 'umax5', 'Y', 'YP', 'a', 'b', 'c']
bounds = [(670, 690), (1000, 1800), (90000, 99000), (0.00001, 0.001), (890, 910), (0.02, 0.06), (0.1, 0.2),
(0.5, 0.7), (0.1, 0.15), (6, 8), (0.8, 1.5), (1.0, 1.9), (0.0001, 0.0015), (0.1, 2)]
```

```
[16]: est, est_info = caretaker.estimate(
    unknowns=unknowns,
    bounds=bounds,
    measurements=data,
    report_level=2,
)
```

-----Results from differential evolution optimizer

```
Estimated parameters:
a: 1.6070606172338695
b: 0.00011617717707478628
c: 1.506590563434198
kace: 1221.8197255173711
kco: 895.6268049854235
kd: 0.03963948329556487
klac: 91871.65862796715
ks: 0.0009339138100783357
S1: 686.2067528664641
umax: 0.133090516154312
umax4: 0.5899235640581024
umax5: 0.10824990347681611
Y: 7.45292016012448
YP: 0.9976335792293202
```

```
[22]: parameters_infos = caretaker.get_parameter_uncertainties(estimates=est, measurements=data, report_level=1)
```

```
Estimated parameters:
-----
a: 1.61e+00 +/- 7.16e-01 (44.56 %)
b: 1.16e-04 +/- 1.26e+00 (1080687.80 %)
c: 1.51e+00 +/- 7.16e+01 (4753.36 %)
kace: 1.22e+03 +/- 1.66e+04 (1358.88 %)
kco: 8.96e+02 +/- 1.86e+04 (2076.46 %)
kd: 3.96e-02 +/- 3.74e-01 (944.25 %)
klac: 9.19e+04 +/- 4.32e+07 (46974.44 %)
ks: 9.34e-04 +/- 9.23e+02 (98822977.17 %)
S1: 6.86e+02 +/- 2.87e+01 (4.19 %)
umax: 1.33e-01 +/- 4.09e-02 (30.74 %)
umax4: 5.90e-01 +/- 1.68e+01 (2848.01 %)
umax5: 1.08e-01 +/- 3.15e+00 (2911.05 %)
Y: 7.45e+00 +/- 8.03e+01 (1077.90 %)
YP: 9.98e-01 +/- 1.08e+01 (1079.26 %)
```

Esta ultima parte del codigo brinda informacion acerca de la fiabilidad de los datos obtenidos en base a los datos brindados, donde implica que mas repeticiones puede brindar mejores datos pero es un problema que requiere equipos de computo costosos, buenos servidores y mucho tiempo para procesarlos. Entre mas pequeña sea la ecuacion, mas facil es de procesar, incluso con equipos de computo domesticos o escolares.



Figura A 31. Gráfico tipo Cornet plot generados por el muestreo de Monte Carlo.

Modelo Estructurado

5.3 Apendice C

Las siguientes tablas se usaron para elaborar el modelado estructurado y la propuesta de mapa metabólico. Las tablas en color verde son relacionadas a la vía metabólica del sulfato a sulfuro y las tablas en azul son relacionadas a la vía metabólica del lactato a acetato.

(Sa) EC 2.7.7.4 sulfato adeniltransferasa				Escherichia coli K12		salvaje / mutante S40A / mutante S35A
Productos		Sustratos		Efecto		Protein data
ATP		difosfato		Modificador-catalizador		complejo proteico
Sulfato		adenililsulfato				((P21156)(P23845))
PARAMETROS						
nombre	tipo	especie	valor inicial	valor final	desviacion	unidades
a	concentración	difosfato	0.15	0.8	*	mM
e	concentración	enzima	0.26	*	*	uM
	Vmax		0.156	*	*	*
b	concentración	adenililsulfato	0.08	0.2	*	mM
kcat	kcat	*	0.6 / 3.2 / 0.82	*	0.008 / 0.3 / 0.003	s ^{^-1}
kia	ki	difosfato	0.098 / 0.45 / 0.18	*	0.008 / 0.08 / 0.12	mM
kib	ki	adenililsulfato	0.2 / 0.12 / 0.14	*	0.02 / 0.02 / 0.07	mM
kmb	km	adenililsulfato	0.1 / 0.49 / 0.13	*	0.005 / 0.08 / 0.05	mM
kma	km	difosfato	0.052 / 1.8 / 0.16	*	0.002 / 0.02 / 0.05	mM
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	25		*		C°	
pH	8		*			
buffer	9.2 units/ml Hexokinase, 0.50 units/ml Glucose-6-phosphate dehydrogenase, 10 mM Glucose, 0.20 mM NADP+, 10 mM KCl, 1.0 mM MgCl2, 50 mM Hepes/K+					
REFERENCIA						
Titulo	Autor	año	journal	volumen	paginas	PubMed
Allosteric and catalytic functions of the PPI-binding motif in the ATP sulfurylase-GTPase system	Pilloff DE, Leyh TS	2003	J Biol Chem	278	50435-41	14506286

<https://sabiork.h-its.org/newSearch?q=ecnumber:1.3.1.110>

(Sar) EC 1.8.99.2 adenilsulfato reductasa				Desulfovibrio gigas NCIB 9332		salvaje
Productos		Sustratos		Efecto		Protein data
Aceptor red. (hexacianoferrato II)		AMP		Modificador-catalizador		complejo proteico
Adenosina 5´ fosfosulfato		Aceptor (hexacianoferrato III)				((T2G6Z9)(T2G899))* n
		sulfito				kDa ((70)()) 400 peso en mol
PARAMETROS						
nombre	tipo	especie	valor inicial	valor final	desviacion	unidades
C	concentracion	hexacianoferrato (III)	1.3 / 1.3	*	*	mM
B	concentracion	Sulfito	3.3 / 0.0	* / 3.3	*	mM
A	concentracion	AMP	0.0 / 3.3	3.3 / *	*	mM
Km	Km	AMP / sulfito	0.16 / 0.34	*		mM
Vmax	Vmax	*	*	*	*	*
ley cinética	M-M	Vmax*A/(Km+A)				
		Vmax*B/(Km+B)				
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	*		*		C°	
pH	7.5		*			
buffer	100 mM Tris/HCl, 5 mM EDTA, 0.5 mM beta-mercaptoethanol					
REFERENCIA						
Titulo	Autor	año	journal	volume n	paginas	PubMed
The active centers of adenylylsulfate reductase from Desulfovibrio gigas. Characterization and spectroscopic studies.	Lampreia J, Moura I, Teixeira M, Peck HD Jr, Legall J, Huynh BH, Moura JJ	1990	Eur J Biochem	188	653-64	2158885

link: <https://sabiork.h-its.org/newSearch?q=ecnumber:1.3.1.110>

<https://www.uniprot.org/uniprotkb/T2G6Z9/entry>

EC 1.8.1.22 Subunidad beta de dipo desasimilador de la sulfito reductasa				Desulfovibrio vulgaris (Hildenborough, NCIB 8303)		Salvaje / dSiR beta
Productos		Sustratos		Efecto		Protein data
DsrC - ditiol		DsrC- trisulfuro		Modificador-catalizador		complejo proteico
sulfito		H2O				
NADH + H		NAD+				varios cofactores unidos a 2 non cubane [4Fe-4S] por subunidad, 1 siroheme por subu
PARAMETROS						
nombre	tipo	especie	valor inicial	valor final	desviacion	unidades
	concentración	*	*	*	*	mM
	concentración	*	*	*	*	mM
	concentración	*	*	*	*	mM
Km	Km	sulfito	0.06 / 0.0155	*		mM
Vmax	Vmax	*	*	*	*	*
	kcat	0.31				s^(-1)
ley cinetica	M-M	Vmax*A/(Km+A)				
		Vmax*B/(Km+B)				
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	298		*		K	
pH	7.6		*			
buffer	Optimal separa-tion of bands I and I1 was obtained by a gradient method,using two stocks (A = 50mM, B = 500 m M) of degassed potassium phosphate pH 7.6. The total running time was 40min (5min 11% B, 17min 11-100% B, and 10min100% B).					
REFERENCIA						
Titulo	Autor	año	journal	volumen	paginas	PubMed
Desulfoviridin, a multimeric-dissimilatory sulfite reductase from Desulfovibrio vulgaris (Hildenborough). Purification, characterization, kinetics and EPR studies	B M Wolfe , S M Lui, J A Cowan	1994	Eur J Biochem	223	79-89	8033912

<https://www.uniprot.org/uniprotkb/P45575/entry>

Parte del complejo que cataliza la reducción de sulfito a sulfuro. Las subunidades alfa y beta pueden haber surgido por duplicación génica. Ambas se unen a dos grupos de hierro y azufre, pero la subunidad alfa parece ser catalíticamente inactiva, debido a sustituciones a lo largo del supuesto canal de acceso al sustrato y porque se une a la sirohidroclorina (la forma desmetilada del sirohemo) en lugar del sirohemo.

dsrA sulfito reductasa, subunidad alfa de tipo disimilatorio

<https://www.uniprot.org/uniprotkb/P45574/entry>

DsrC-sulfurotransferasa

<https://www.uniprot.org/uniprotkb/Q87899/entry>

EC 2.3.1.8 Fosfato acetil transferasa				Escherichia coli K12 AG1		salvaje N-terminal His-tag
Productos		Sustratos		Efecto		Protein data
Fosfato		Coenzima A		Modificador-catalizador		complejo proteico
Acetil-CoA		Acetil fosfato				(P0A9M8)*6
PARAMETROS						
nombre	tipo	especie	valor inicial	valor final	Desviación estandar	unidades
A	concentración	Coenzima A / Acetil CoA	0.2 / 0.1	*	*	mM
S	concentración	Acetil fosfato / Fosfato	*	*	*	*
Kcat	Kcat	*	227.6 / 29.6	*	9.3 / 2.3	s^(-1)
Km	Km	Acetil fosfato / Fosfato	0.9 / 2.1	*	0.1 / 0.2	mM
Vmax	Vmax	*	177.4 / 23.1	*	6.2 / 2.1	U/mg
ley cinética	M-M	Vmax*S/(Km+S)				
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	30		*		C°	
pH	*		*			
buffer	50 mM Tris-HCl, 20 mM KCl, 2 mM Dithiothreitol					
REFERENCIA						
Titulo	Autor	año	journal	volumen	paginas	PubMed
Functional dissection of Escherichia coli phosphotransacetylase structural domains and analysis of key compounds involved in activity regulation	Campos-Bermudez VA, Bologna FP, Andreo CS, Drincovich MF	2010	FEBS J	277	1957-66	20236319

EC 2.7.2.1 Acetato quinasa				Escherichia coli		salvaje
Productos		Sustratos		Efecto		Protein data
Acetato		ADP		Modificador-catalizador		complejo proteico
ATP		Acetil fosfato				(P0A6A3)*2;
PARAMETROS						
nombre	tipo	especie	valor inicial	valor final	Desviación estandar	unidades
A	concentración	ADP	0.2	1.0	*	mM
E	concentración	Enzima	0.5	*	*	ug/ml
B	concentración	Acetil fosfato / Fosfato	0.14	1.0	*	mM
Kib	Kd	Acetil fosfato	0.14	*	0.020	uM
Kia	Kd	ADP	50.0	*	20	uM
Kb	Km	Acetil fosfato	0.34		0.04	uM
Ka	Km	ADP	360.0		100	uM
V	Vmax	*	68.0		3	U/mg
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	25		*		C°	
pH	7		*		*	
buffer	50 mM Piperazine-N,N'-bis(2-ethanesulfonic acid), 3 mM Mg2+, 0.3 mM NAD+, 2 mM glucose, 33.3 µg/ml hexokinase, 13.3 µg/ml glucose-6-phosphate dehydrogenase					
REFERENCIA						
Titulo	Autor	año	journal	volumen	paginas	PubMed
The inhibition of acetate, pyruvate, and 3-phosphoglycerate kinases by chromium adenosine triphosphate	Janson CA, Cleland WW	1970	J Biol Chem	249	2567-71	4362687

EC 1.2.7.1 Piruvato:ferredoxina oxidorreductasa				Moorella thermoacetica Desulfovibrio africanus		
Productos		Sustratos		Efecto		Protein data
Acetil CoA		CoA		Cataliza la descarboxilación oxidativa del piruvato a acetil-CoA y dióxido de carbono.		varios sitios de cofactores
CO2		Piruvato		*		Se une a 3 grupos [4Fe-4S] por subunidad.Se une a 1 pirofosfato de tiamina por subunidad. Se une a 1 ion Mg 2+ por subunidad.
2 reducido [2Fe-2S]-ferredoxina + H		2 oxidado [2Fe-2S]-ferredoxina		*		*
PARAMETROS						
tipo	sustrato	valor inicial	unidades	tipo	Función	unidades
Km	Acetil CoA	9	uM	Vmax	para la síntesis de piruvato a partir de acetil-CoA y CO2 con metilviológeno como donador de electrones	1,6 µmol/min/mg
Km	CO2	2	mM	Vmax	para la síntesis de piruvato con ferredoxina reducida como donador de electrones	1 µmol/min/mg
Km	Piruvato	0.3	mM	Vmax	para la descarboxilación oxidativa del piruvato con ferredoxina oxidada como aceptor de electrones	14,8 µmol/min/mg
Km	Ferredoxina oxidada	0.32	uM	*	*	*
Km	Ferredoxina reducida	0.27	uM	kcat	para la síntesis de piruvato a partir de acetil-CoA y CO2 con ferredoxina reducida como donador de electrones	2.0 s ⁽⁻¹⁾
*	*	*	*	kcat	para la descarboxilación oxidativa del piruvato con ferredoxina oxidada como aceptor de electrones	29.6 s ⁽⁻¹⁾
*	*	*	*	kcat	para la descarboxilación oxidativa del piruvato con rubredoxina oxidada como aceptor de electrones	19.8 s ⁽⁻¹⁾
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	*		*		C°	
pH	*		*			
buffer	*					
REFERENCIA						
Titulo	Autor	año	journal	volumen	paginas	PubMed
The role of pyruvate ferredoxin oxidoreductase in pyruvate	Furdui C., Ragsdale S.W.	2000	J Biol Chem	275(37)	28494-28499	10878009

synthesis during autotrophic growth by the Wood-Ljungdahl pathway.						
--	--	--	--	--	--	--

<https://www.uniprot.org/uniprotkb/Q2RMD6/entry>

<https://doi.org/10.1074/jbc.m003291200>

EC 1.1.5.12 D lactato deshidrogenasa				Escherichia coli K12		
Productos		Sustratos		Efecto		Protein data
Piruvato		(R)-Lactato		Cataliza la oxidación del D-lactato a piruvato. Los electrones derivados de la oxidación del D-lactato se transfieren a la cadena de transferencia de electrones de la ubiquinona/citocromo, donde pueden utilizarse para proporcionar energía para el transporte activo de una variedad de aminoácidos y azúcares a través de la membrana.		varios sitios de cofactores
Quinol		Quinona				Se une a 3 grupos [4Fe-4S] por subunidad.Se une a 1 pirofosfato de tiamina por subunidad. Se une a 1 ion Mg 2+ por subunidad.
PARAMETROS						
tipo	sustrato	valor inicial	unidades	tipo	valor	unidades
Km	D-lactato	0.16	mM	Vmax	4.32	mM/s
Km			uM	kcat	0.36	s^(-1)
CONDICIONES EXPERIMENTALES						
	valor inicial		valor final		unidades	
temperatura	25		*		C°	
pH	8.2 / 7.1		*			
buffer	The samples were prepared by adding 10 /xl of 2.3 mM D-LDH to 500 /~l of solution containing 25 mM D-lactate, L-lactate, or pyruvate in 0.1 M potassium phosphate buffer in D20 at pH 8.2, under anaerobic conditions. D-lactate, L-lactate, or pyruvate in 0.1 M potassium phosphate buffer in D20 at pH 8.2					
REFERENCIA						
Titulo	Autor	año	journal	volumen	paginas	PubMed
Stopped-flow kinetic and biophysical studies of membrane-associated D-lactate dehydrogenase of Escherichia coli.	Furdui C., Ragsdale S.W.	1995	Biochimica et Biophysica Acta	1252	269-277	7578233

<https://www.uniprot.org/uniprotkb/P06149/entry>

5.4 Apéndice D

Código 5: sensibilidad de parámetros para el modelo estructurado

```
[4]: # importar desde el paquete de pyfoomb
import pyfoomb
print(f'Current package version of {pyfoomb.__name__}: {pyfoomb.__version__}')

from pyfoomb import BioprocessModel
from pyfoomb import ObservationFunction
from pyfoomb import Caretaker
from pyfoomb import Visualization
from pyfoomb import Measurement

Current package version of pyfoomb: 2.17.7

[6]: from matplotlib import pyplot
import numpy
import pandas

# Set seed of random number generator for reproducibility
numpy.random.seed(123)

[8]: # Define la clase de modelo
class SequentialKinetic(BioprocessModel):

    def rhs(self, t, y):
        # Descomprime el vector de estado. Los estados son ordenados alfabeticamente. #pyx11 para extraeer datos
        A, B, C, D, E, F, G, L, P, S, X = y

        # Descomprime los parámetros del modelo. Ejemplo:
        S1 = self.model_parameters['S1']
        kace = self.model_parameters['kace']
        klac = self.model_parameters['klac']
        ks = self.model_parameters['ks']
        kco = self.model_parameters['kco']
        kd = self.model_parameters['kd']
        #velocidades
        umax = self.model_parameters['umax']
        umax3 = self.model_parameters['umax3']
        umax5 = self.model_parameters['umax5']
        #rendimientos
        Y = self.model_parameters['Y']
        YP = self.model_parameters['YP']
        #exponentes
        a = self.model_parameters['a']
        b = self.model_parameters['b']
        c = self.model_parameters['c']
```

```

#ruta metabolica
umax1 = self.model_parameters['umax1']
kb = self.model_parameters['kb']
umax2 = self.model_parameters['umax2']
kd2 = self.model_parameters['kd2']
umax4 = self.model_parameters['umax4']
ke = self.model_parameters['ke']
umax6 = self.model_parameters['umax6']
kf = self.model_parameters['kf']
umax7 = self.model_parameters['umax7']
kg = self.model_parameters['kg']

# Define las derivadas.
#Sulfato
dSdt = -(Y)*(umax)*((1-(P/S1)**a)*(S/(ks+S)))*X*(L**b)
dBdt = ((umax1)*((B*S)/(kb+B)))*(X)*(C/(kco+ C))
dDdt = ((umax2)*((D*B)/(kd2+D+(D**2)))*(X)*(C/(kco+ C))
dPdt = (umax*(YP))*(((1-(P/S1)**a)*(S/(ks+S)))*X*(L**b)
#Lactato
dLdt = (-umax3)*((L**c)/(klac+L**c))*X*(kace/(A+kace))
dEdt = ((umax4)*((E*L)/(ke+E)))*(X)*(C/(kco+ C))
dFdt = ((umax6)*((F*E)/(kf+F)))*(X)*(C/(kco+ C))
dGdt = ((umax7)*((G*F)/(kg+G)))*(X)*(C/(kco+ C))
dAdt = (umax3)*((L**c)/(klac+L**c))*X*(kace/(A+kace))
#Carbono
dCdt = (umax5)*((L**c)/(klac+L**c))*X*(kco/(kco+C))

#Biomasa
dXdt = (umax*0.6)*((1-(P/S1)**a)*(S/(ks+S))*X)-((kd*X*(S**b))/5)

# Regresa las derivadas como Lista (or numpy array)
# El orden corresponde al vector de estado.
return [dAdt, dBdt, dCdt, dDdt, dEdt, dFdt, dGdt, dLdt, dPdt, dSdt, dXdt]

# Define un diccionario para los valores iniciales.
# La clave corresponde al estado del modelo, extendidos por un 0 (cero).
#
initial_values = {
    'A0':0.1,
    'B0':0.1,
    'C0':10,
    'D0':0.1,
    'E0':0.1,
    'F0':0.01,
    'G0':0.0001,
    'L0':4640,
    'P0':29,
    'S0':5655,
    'X0':5,
}

```

```

# Define un diccionario para los parametros del modelo.
# La clave se relaciona con los nombres de las variables utilizadas en la clase del modelo. Ejemplo:
model_parameters = {
    #constantes cinéticas
    'S1':680,
    'kace':1500,
    'klac':9500,
    'ks':0.0009048,
    'kco':9,
    'kd':0.04,
    #velocidades
    'umax':0.13,
    'umax3':0.6,
    'umax5':0.12,
    #rendimientos
    'Y':7.5,
    'YP':1,
    #exponentes
    'a':1.9,
    'b':0.1,
    'c':1.5,

    #ruta metabolica
    'kb':0.036,
    'umax1':0.000056,
    'kd2':0.12,
    'umax2':0.0055,
    'ke':0.011,
    'umax4':0.000078,
    'kf':0.017,
    'umax6':0.00015,
    'kg':0.00012,
    'umax7':0.00038,
}

# Instanciar un Caretaker, que administra el modelo y su parametrización.
caretaker = Caretaker(
    bioprocess_model_class=SequentialKinetic,
    model_parameters=model_parameters,
    initial_values=initial_values,
)

# Ejecuta una simulación hacia adelante.
simulation = caretaker.simulate(t=216)

# Visualizar las trayectorias resultantes.
_ = Visualization.show_kinetic_data(simulation)

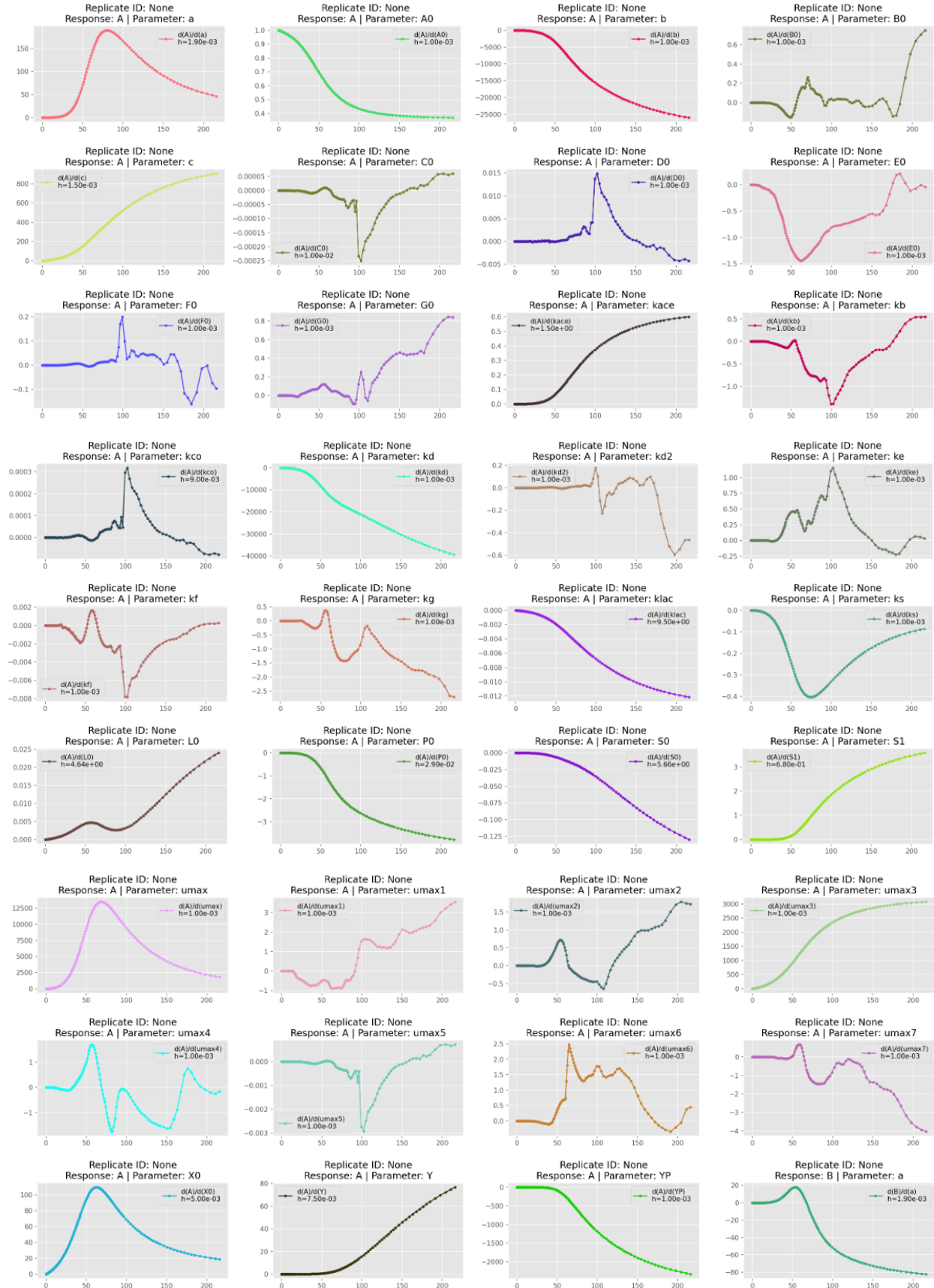
```

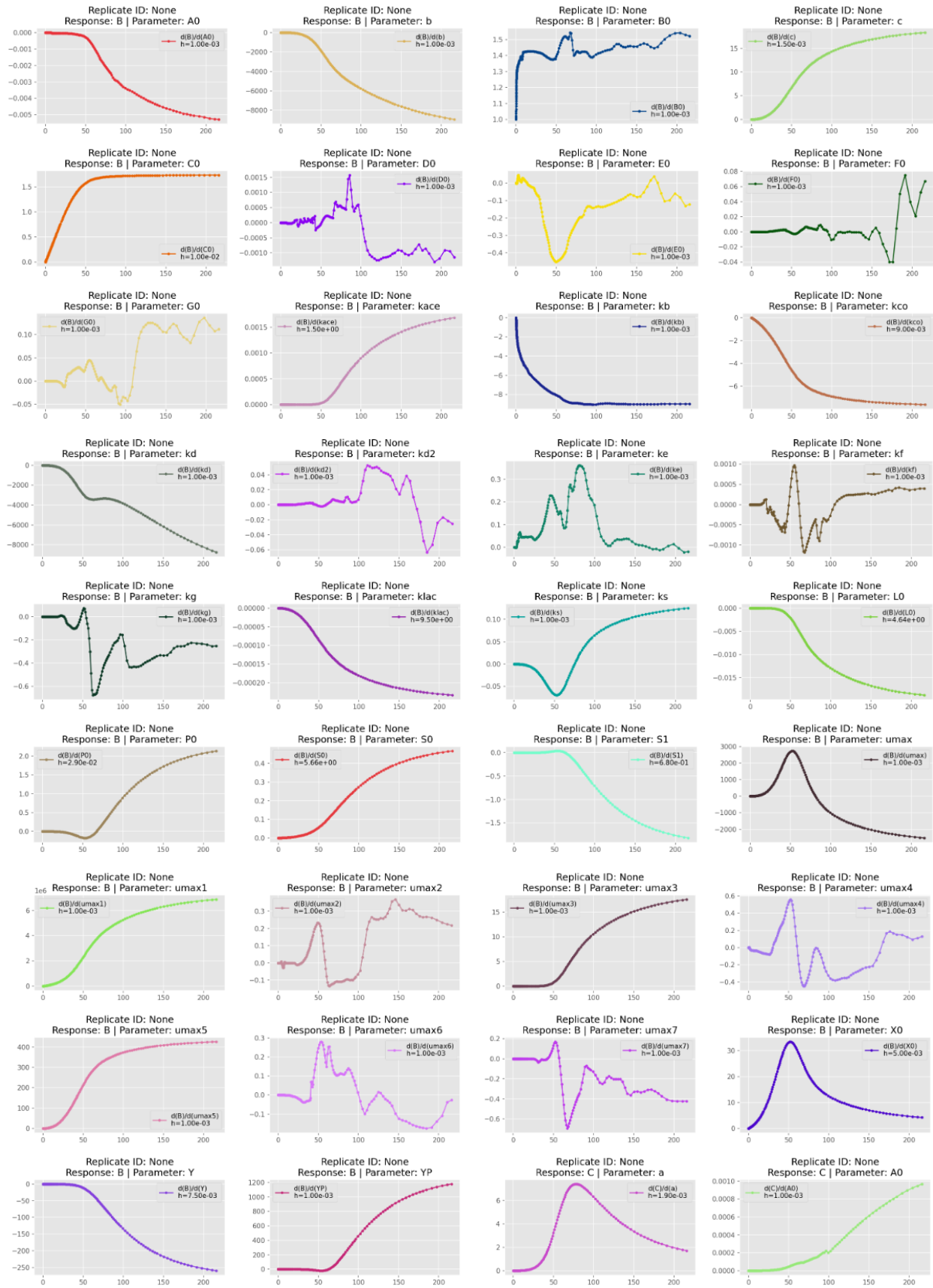
Figura A 32. Resultados del código para el análisis de sensibilidad del modelo estructurado.

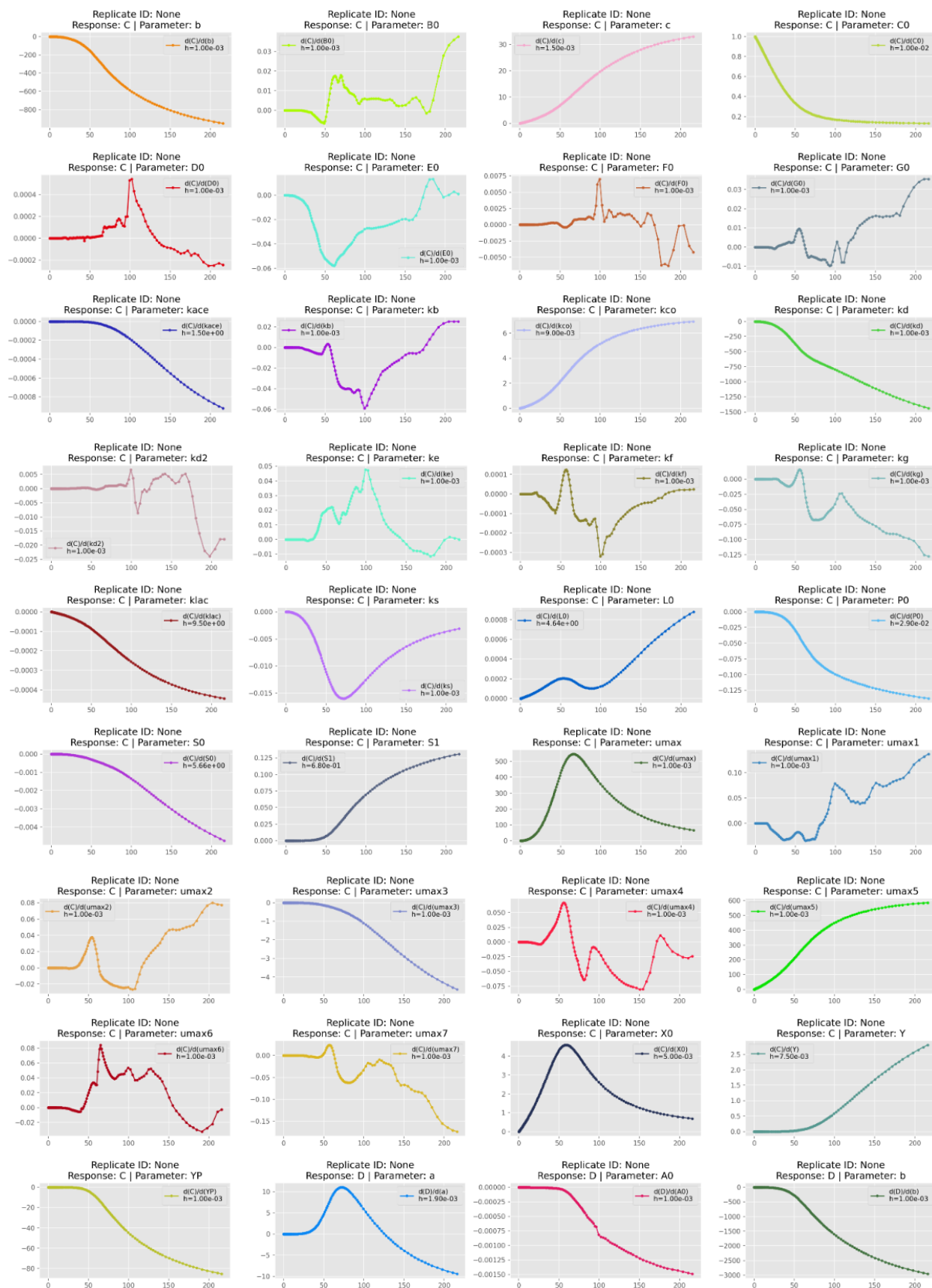
```

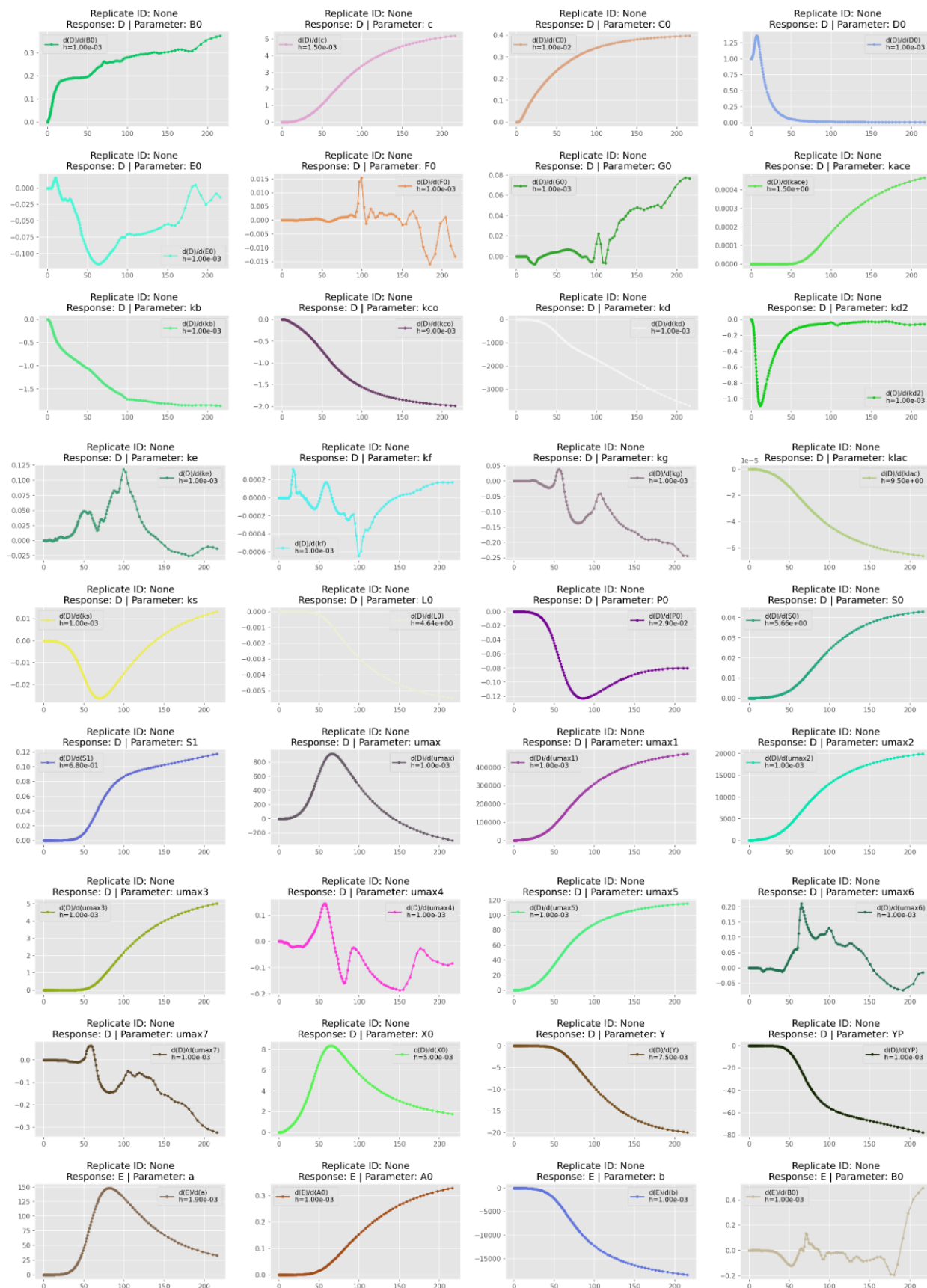
[50]: # Obtenga todas las sensibilidades y muéstrelas.
      # NOTA: Esto generará un gráfico enorme
      sensis_all = caretaker.get_sensitivities(tfinal=216)
      _ = Visualization.show_kinetic_data(sensis_all, ncols=4)

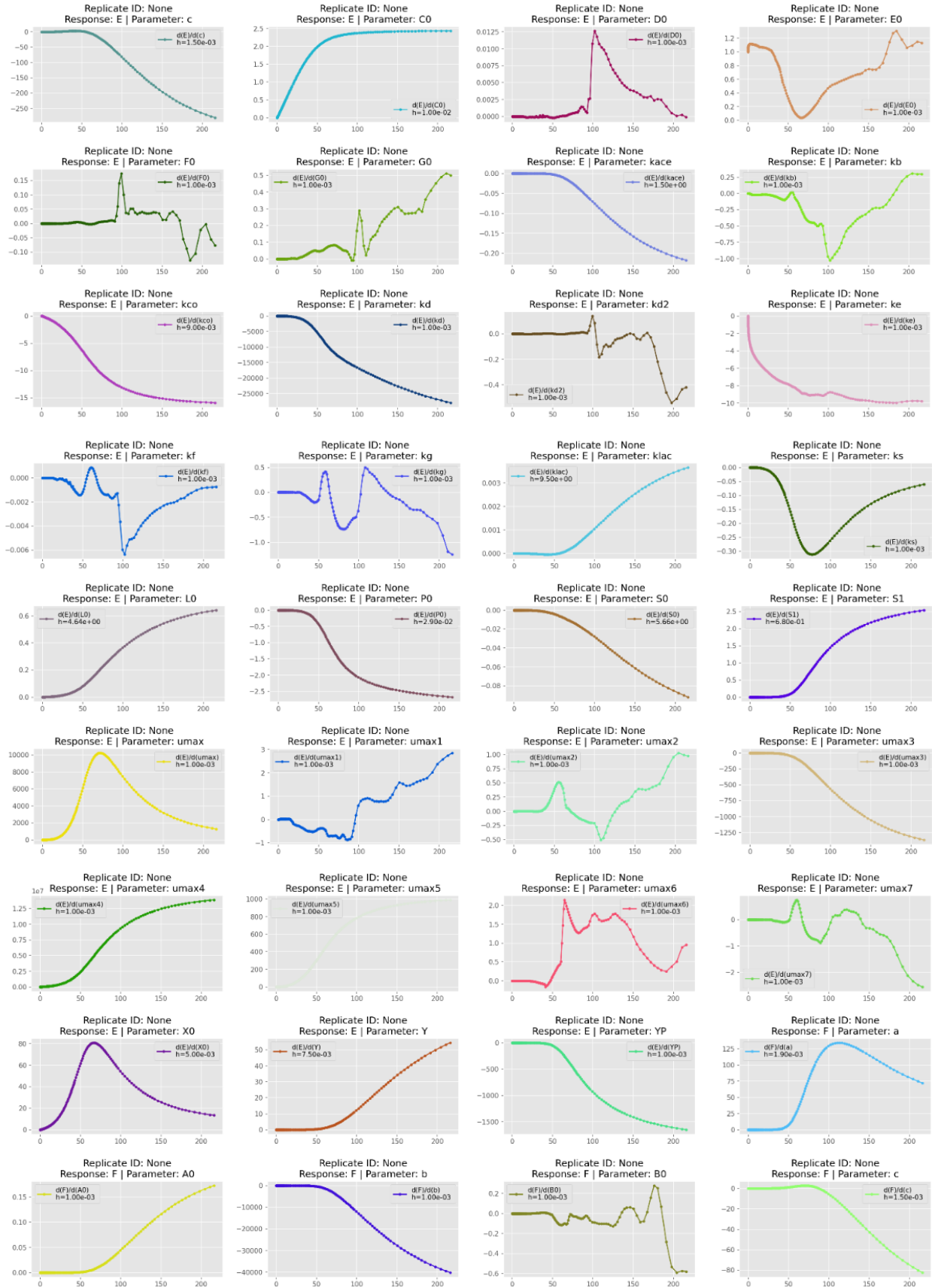
```

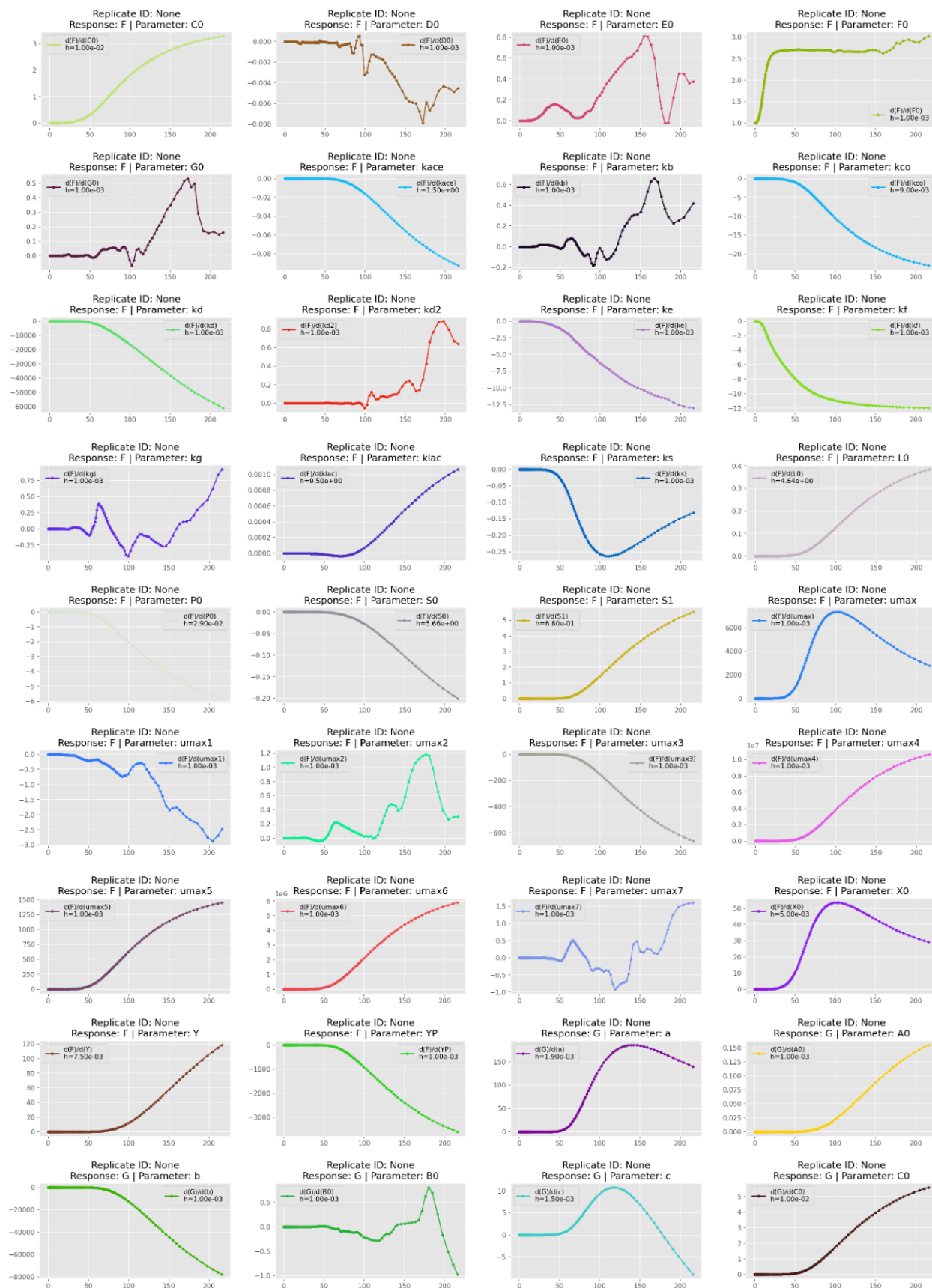


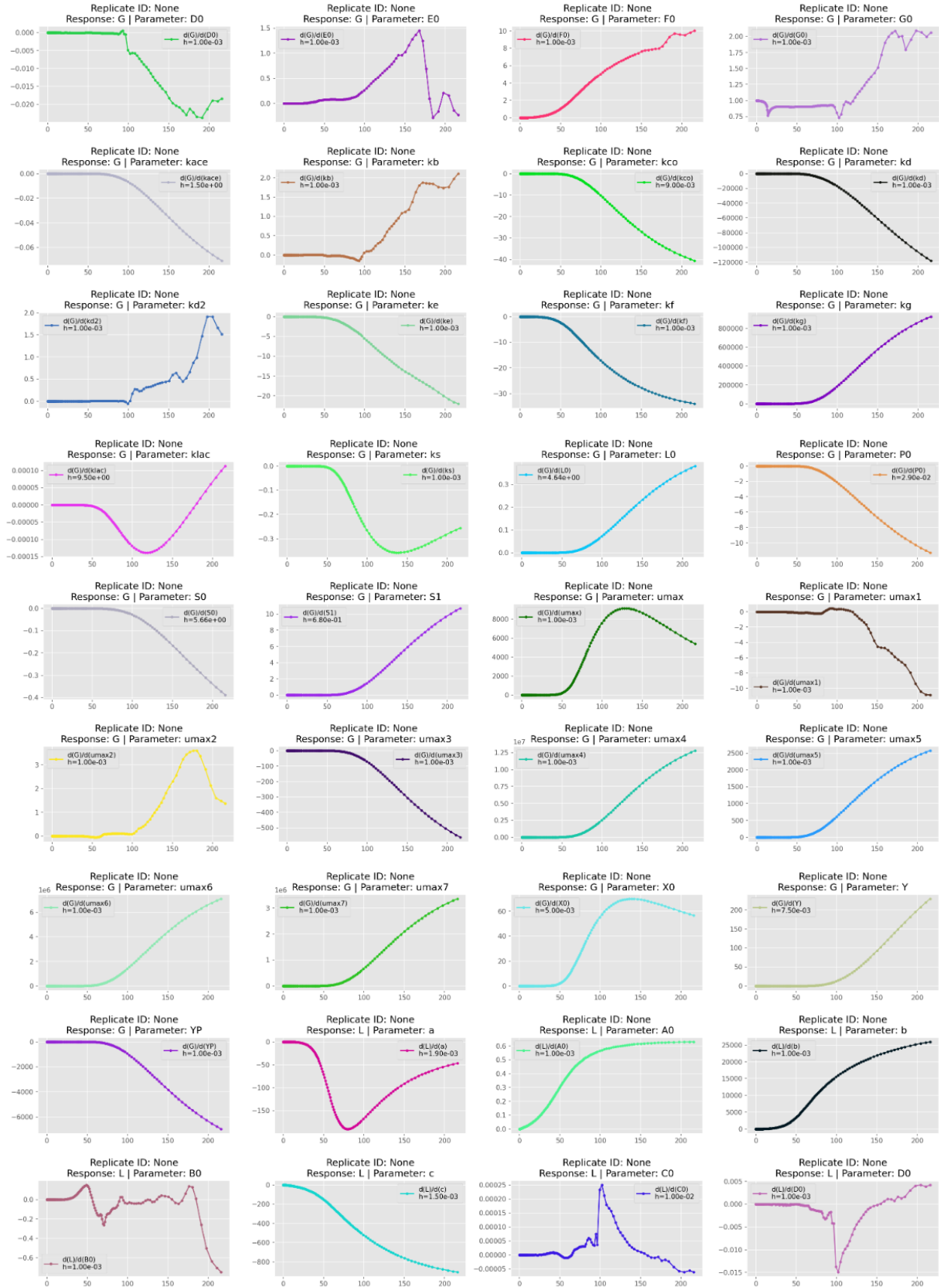


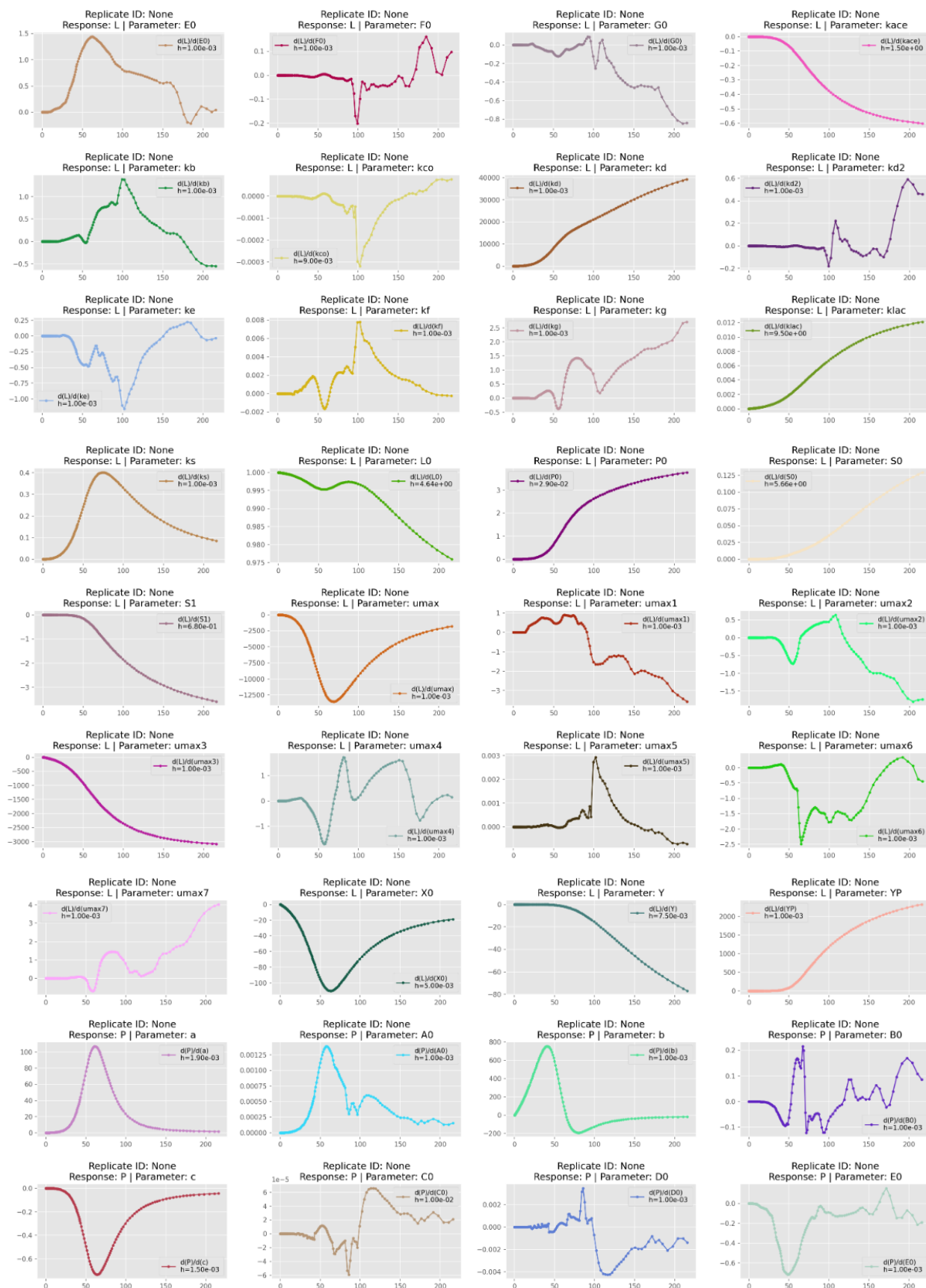


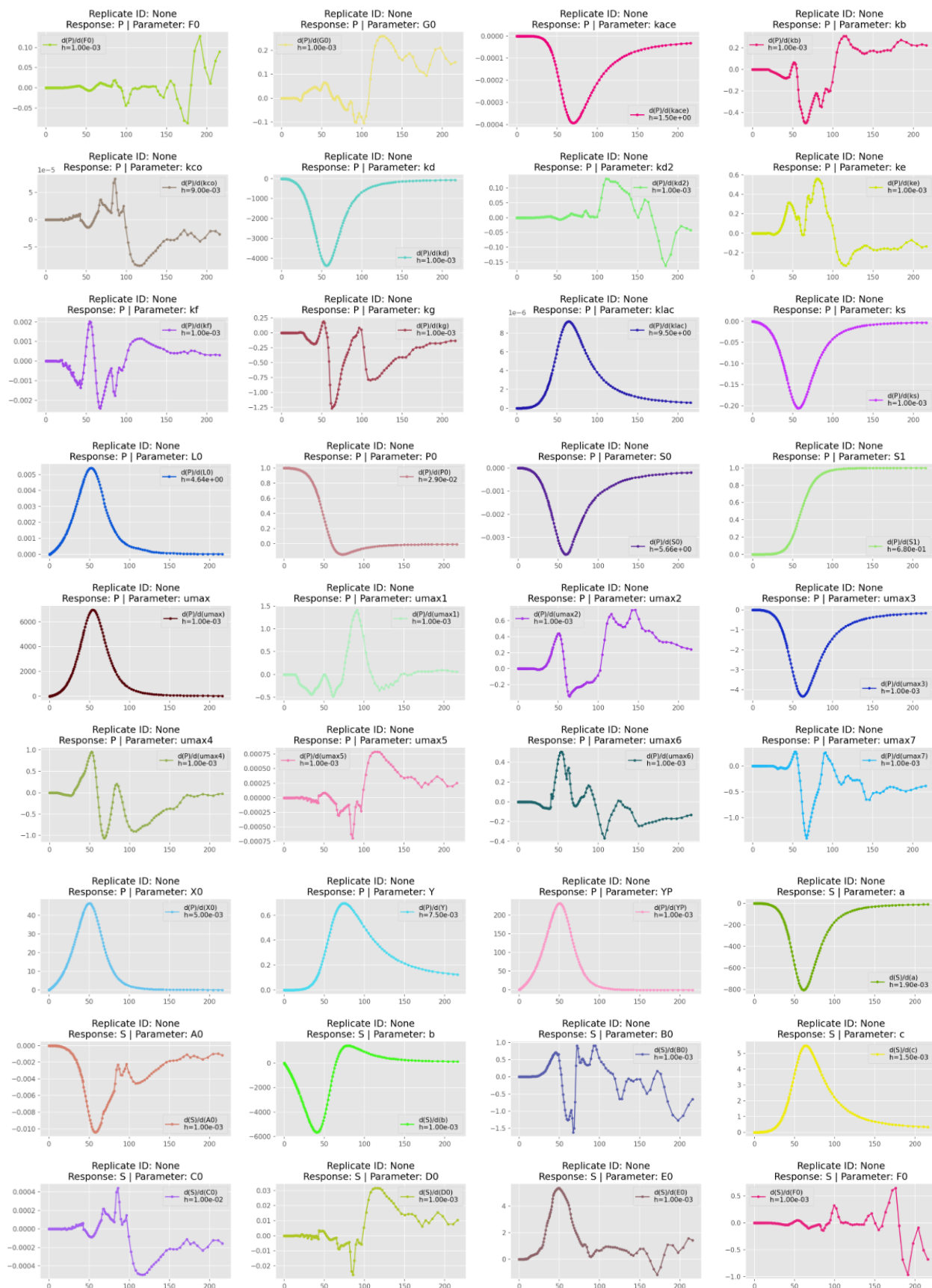


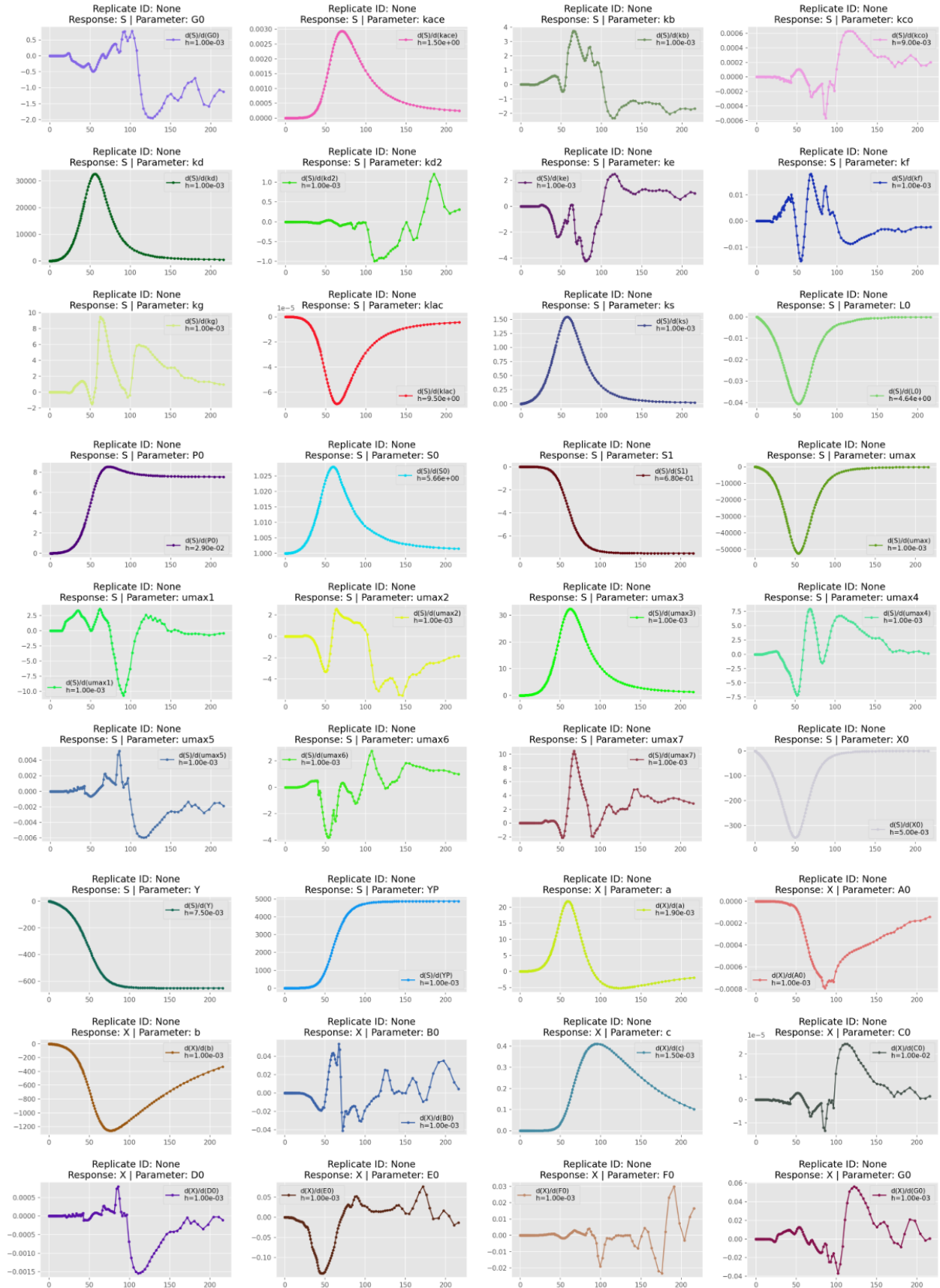


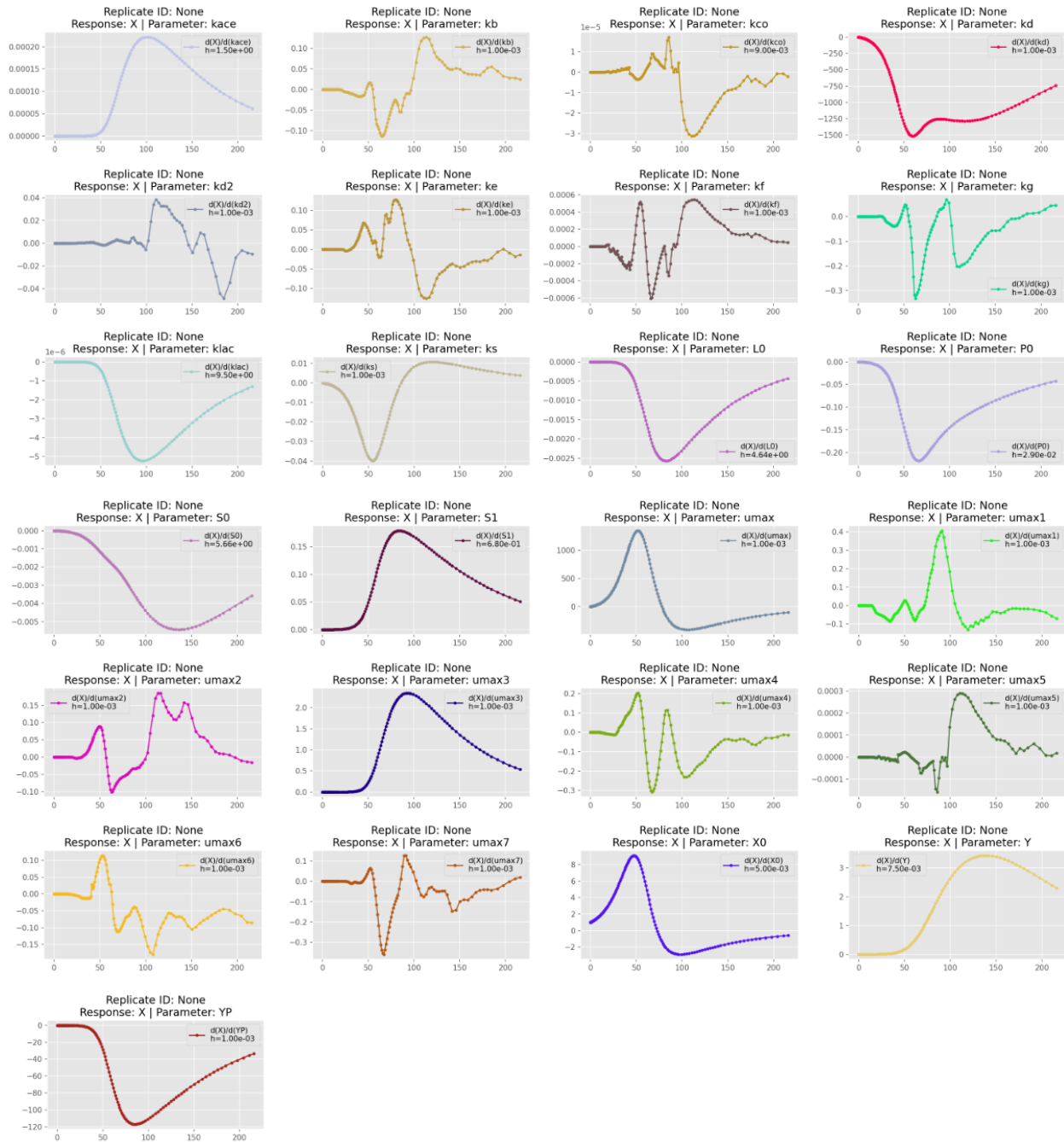












Esta ultima parte del codigo permite vizualizar todos los parametros de cualquier ecuacion entre si para saber quien influye positiva o negativamente en determinados puntos de las graficas.

Código 6 en Rstudio

Código para la creación del mapa de calor a partir de la tabla de sensibilidades con valores numericos del modelo expandido para la obtencion de ña figura 22 de la seccion de resultados.

```
1 library(pheatmap)
2 library(readxl)
3
4 datos_rcompleta <- as.data.frame(Sensibilidad_ruta_meta_completa)
5 rownames(datos_rcompleta) <- datos_rcompleta[,1]
6 datos_rcompleta <- datos_rcompleta[,-1]
7 str(datos_rcompleta)
8
9
10
11 pheatmap(datos_rcompleta,
12          main = "Sensibilidad entre variables",
13          display_numbers = FALSE,
14          fontsize_row = 6,
15          fontsize_col = 7,
16          breaks = c(0,1,2,3,4,5),
17          legend_breaks = 1:5,
18          border_color = "gray",
19          cellheight = 8,
20          cellwidth = 20,
21          treeheight_row = 15,
22          cluster_rows = TRUE,
23          cluster_cols = TRUE,
24          color = colorRampPalette(c("yellow", "orange", "red"))(5))
25
```

5.5 Apéndice E

Código 7 análisis de sensibilidad del modelo expandido y generación de los gráficos y tablas de relevancia paramétrica.

```
# importamos desde el paquete de pyfoomb
import pyfoomb
print(f'Current package version of {pyfoomb.__name__}: {pyfoomb.__version__}')

from pyfoomb import BioprocessModel
from pyfoomb import Caretaker
from pyfoomb import Visualization
```

Current package version of pyfoomb: 2.17.7

```
# Authors: The scikit-Learn developers
# SPDX-License-Identifier: BSD-3-Clause

import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from sklearn import ensemble
from sklearn.inspection import permutation_importance
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.utils.fixes import parse_version
```

```
class SequentialKinetic(BioprocessModel):

    def rhs(self, t, y):
        A, B, C, D, E, F, G, L, P, S, X = y

        # Parámetros (usar Los ajustados de arriba)
        S1 = self.model_parameters['S1']
        kace = self.model_parameters['kace']
        klac = self.model_parameters['klac']
        ks = self.model_parameters['ks']
        kco = self.model_parameters['kco']
        kd = self.model_parameters['kd']
        umax = self.model_parameters['umax']
        umax3 = self.model_parameters['umax3']
        umax5 = self.model_parameters['umax5']
        Y = self.model_parameters['Y']
        YP = self.model_parameters['YP']
        a = self.model_parameters['a']
        b = self.model_parameters['b']
        c = self.model_parameters['c']
        umax1 = self.model_parameters['umax1']
        kb = self.model_parameters['kb']
        umax2 = self.model_parameters['umax2']
        kd2 = self.model_parameters['kd2']
        umax4 = self.model_parameters['umax4']
        ke = self.model_parameters['ke']
        umax6 = self.model_parameters['umax6']
        kf = self.model_parameters['kf']
        umax7 = self.model_parameters['umax7']
        kg = self.model_parameters['kg']
```

```

# PROTECCIÓN: Evitar valores negativos en variables clave
S = max(S, 0.001) # Mínimo de sustrato
X = max(X, 0.001) # Mínimo de biomasa
P = max(P, 0.001) # Mínimo de producto

# Término de inhibición por producto - protegido
inhibition_term = max(0.001, 1 - (P/S1)**a)

# Término de Monod para sustrato - protegido
monod_term = S / (ks + S)

# ECUACIÓN MODIFICADA con protección completa
growth_term = (umax) * inhibition_term * monod_term * X
death_term = (kd * X * (S**b)) / 5

# Asegurar que death_term no sea mayor que growth_term
death_term = min(death_term, growth_term * 0.8) # Máximo 80% del crecimiento

dXdt = growth_term - death_term

# Resto de ecuaciones (también con protecciones)
dSdt = -(Y) * (umax) * inhibition_term * monod_term * X * (L**b)
dBdt = ((umax1) * ((B * S) / (kb + B))) * X * (C / (kco + C))
dDdt = ((umax2) * ((D * B) / (kd2 + D + (D**2)))) * X * (C / (kco + C))
dPdt = (umax * (YP)) * inhibition_term * monod_term * X * (L**b)
dLdt = (-umax3) * ((L**c) / (klac + L**c)) * X * (kace / (A + kace))
dEdt = ((umax4) * ((E * L) / (ke + E))) * X * (C / (kco + C))
dFdt = ((umax6) * ((F * E) / (kf + F))) * X * (C / (kco + C))
dGdt = ((umax7) * ((G * F) / (kg + G))) * X * (C / (kco + C))
dAdt = (umax3) * ((L**c) / (klac + L**c)) * X * (kace / (A + kace))
dCdt = (umax5) * ((L**c) / (klac + L**c)) * X * (kco / (kco + C))

return [dAdt, dBdt, dCdt, dDdt, dEdt, dFdt, dGdt, dLdt, dPdt, dSdt, dXdt]

```

```

# Define un diccionario para los valores iniciales.
# La clave corresponde al estado del modelo, extendidos por un 0 (cero).
#
initial_values = {
    'A0':0.1,
    'B0':0.1,
    'C0':10,
    'D0':0.1,
    'E0':0.1,
    'F0':0.01,
    'G0':0.0001,
    'L0':4640,
    'P0':29,
    'S0':5655,
    'X0':5,
}

```

```

# Define un diccionario para los parametros del modelo.
# La clave se relaciona con los nombres de las variables utilizadas en la clase del modelo. Ejemplo:
model_parameters = {
    #constantes cinéticas
    'S1':680,
    'kace':1500,
    'klac':9500,
    'ks':0.0009048,
    'kco':9,
    'kd':0.04,
    #velocidades
    'umax':0.13,
    'umax3':0.6,
    'umax5':0.12,
    #rendimientos
    'Y':7.5,
    'YP':1,
    #exponentes
    'a':1.9,
    'b':0.1,
    'c':1.5,

    #ruta metabolica
    'kb':0.036,
    'umax1':0.000056,
    'kd2':0.12,
    'umax2':0.0055,
    'ke':0.011,
    'umax4':0.000078,
    'kf':0.017,
    'umax6':0.00015,
    'kg':0.00012,
    'umax7':0.00038,
}

```

```

# Instanciar un Caretaker, que administra el modelo y su parametrización.
caretaker = Caretaker(
    bioprocess_model_class=SequentialKinetic,
    model_parameters=model_parameters,
    initial_values=initial_values,
)

# Ejecuta una simulación hacia adelante.
simulation = caretaker.simulate(t=216)

# Visualizar las trayectorias resultantes.
_ = Visualization.show_kinetic_data(simulation)

```

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.preprocessing import StandardScaler

```

```
def generate_complete_dataset(n_samples=200):
    """
    Genera dataset con todas las variables de estado como targets
    """
    variable_params = [
        'S1', 'kace', 'klac', 'ks', 'kco', 'kd', 'umax',
        'umax3', 'umax5', 'Y', 'YP', 'a', 'b', 'c',
        'kb', 'umax1', 'kd2', 'umax2', 'ke', 'umax4', 'kf',
        'umax6', 'kg', 'umax7'
    ]

    # Diccionarios para almacenar datos
    features_dict = {param: [] for param in variable_params}
    targets_dict = {state: [] for state in ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'L', 'P', 'S', 'X']}
    trajectories_dict = {state: [] for state in ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'L', 'P', 'S', 'X']}

    successful_simulations = 0

    for i in range(n_samples):
        try:
            # Variar parámetros
            varied_params = model_parameters_stable.copy()
            for param in variable_params:
                if param in model_parameters_stable:
                    variation = np.random.uniform(0.8, 1.2)
                    varied_params[param] = model_parameters_stable[param] * variation
```

```
        # Simulación
        temp_caretaker = Caretaker(
            bioprocess_model_class=SequentialKinetic,
            model_parameters=varied_params,
            initial_values=initial_values,
        )

        simulation = temp_caretaker.simulate(t=100)

        # Extraer valores finales de TODAS las variables
        valid_simulation = True
        final_values = {}
        trajectory_data = {}

        for state_name in targets_dict.keys():
            values = extract_simulation_values(simulation, state_name)
            final_val = values[-1]

            # Verificar que los valores sean válidos
            if np.isnan(final_val) or final_val < 0:
                valid_simulation = False
                break

            final_values[state_name] = final_val
            trajectory_data[state_name] = values
```

```
    if valid_simulation:
        # Almacenar características
        for param in variable_params:
            features_dict[param].append(varied_params[param])

        # Almacenar targets
        for state_name, value in final_values.items():
            targets_dict[state_name].append(value)
            trajectories_dict[state_name].append(trajectory_data[state_name])

        successful_simulations += 1

    except Exception as e:
        continue

    print(f"Simulaciones exitosas: {successful_simulations}/{n_samples}")

    # Convertir a arrays
    X = np.column_stack([features_dict[param] for param in variable_params])
    y_dict = {state: np.array(targets_dict[state]) for state in targets_dict.keys()}

    return X, y_dict, variable_params, trajectories_dict
```

```
def analyze_all_variables_sensitivity(X, y_dict, feature_names):
    """
    Realiza análisis de sensibilidad para cada variable de estado
    """
    print("=" * 70)
    print("ANÁLISIS DE SENSIBILIDAD - TODAS LAS VARIABLES")
    print("=" * 70)

    results = {}

    for state_name, y_values in y_dict.items():
        print(f"\n🔍 Analizando variable: {state_name}")
        print("-" * 40)

        # Verificar que hay variabilidad en los datos
        if np.std(y_values) < 1e-10:
            print(f"⚠️ No hay variabilidad en {state_name}, saltando...")
            continue

        # Dividir datos
        X_train, X_test, y_train, y_test = train_test_split(
            X, y_values, test_size=0.2, random_state=42
        )

        # Entrenar modelo
        rf = RandomForestRegressor(n_estimators=100, random_state=42)
        rf.fit(X_train, y_train)

        # Evaluar
        y_pred = rf.predict(X_test)
        r2 = r2_score(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)
```

```
print(f" R²: {r2:.4f}")
print(f" MSE: {mse:.2f}")

# Importancia de características
importance = rf.feature_importances_
importance_df = pd.DataFrame({
    'feature': feature_names,
    'importance': importance
}).sort_values('importance', ascending=False)

# Almacenar resultados
results[state_name] = {
    'r2': r2,
    'mse': mse,
    'importance_df': importance_df,
    'model': rf
}

# Mostrar top 3 parámetros
top_3 = importance_df.head(3)
print(" Top 3 parámetros importantes:")
for _, row in top_3.iterrows():
    print(f" • {row['feature']}: {row['importance']:.4f}")

return results
```

```
def create_comparative_visualization(results, feature_names):
    """
    Crea visualizaciones comparativas para todas las variables
    """
    # 3.1 Heatmap de importancias
    importance_matrix = np.zeros((len(results), len(feature_names)))

    fig, axes = plt.subplots(2, 2, figsize=(20, 15))

    # Heatmap
    state_names = list(results.keys())
    for i, state_name in enumerate(state_names):
        importance_df = results[state_name]['importance_df']
        for j, feature in enumerate(feature_names):
            if feature in importance_df['feature'].values:
                importance_matrix[i, j] = importance_df[importance_df['feature'] == feature]['importance'].values[0]

    # Plot heatmap
    im = axes[0, 0].imshow(importance_matrix, cmap='YlOrRd', aspect='auto')
    axes[0, 0].set_xticks(range(len(feature_names)))
    axes[0, 0].set_xticklabels(feature_names, rotation=45, ha='right')
    axes[0, 0].set_yticks(range(len(state_names)))
    axes[0, 0].set_yticklabels(state_names)
    axes[0, 0].set_title('Importancia de Parámetros por Variable')
    plt.colorbar(im, ax=axes[0, 0], label='Importancia')

    # 3.2 R² por variable
    r2_values = [results[state]['r2'] for state in state_names]
    bars = axes[0, 1].bar(range(len(state_names)), r2_values, color='skyblue', alpha=0.7)
    axes[0, 1].set_xticks(range(len(state_names)))
    axes[0, 1].set_xticklabels(state_names, rotation=45, ha='right')
    axes[0, 1].set_title('R² por Variable de Estado')
    axes[0, 1].set_ylabel('R² Score')
    axes[0, 1].set_ylim(0, 1)

    # Añadir valores en las barras
    for bar, r2 in zip(bars, r2_values):
        height = bar.get_height()
        axes[0, 1].text(bar.get_x() + bar.get_width()/2., height + 0.01,
                        f'{r2:.3f}', ha='center', va='bottom', fontsize=9)

    # 3.3 Top parámetros globales
    global_importance = pd.DataFrame({
        'feature': feature_names,
        'total_importance': np.sum(importance_matrix, axis=0)
    }).sort_values('total_importance', ascending=False)

    top_global = global_importance.head(10)
    axes[1, 0].barh(top_global['feature'], top_global['total_importance'], color='lightgreen')
    axes[1, 0].set_title('Importancia Global de Parámetros (Suma sobre todas las variables)')
    axes[1, 0].set_xlabel('Importancia Acumulada')

    # 3.4 Especificidad de parámetros
    specificity = np.std(importance_matrix, axis=0) / (np.mean(importance_matrix, axis=0) + 1e-10)
    spec_df = pd.DataFrame({
        'feature': feature_names,
        'specificity': specificity
    }).sort_values('specificity', ascending=False)

    top_specific = spec_df.head(8)
    axes[1, 1].barh(top_specific['feature'], top_specific['specificity'], color='orange')
    axes[1, 1].set_title('Especificidad de Parámetros\n(Alta = afecta pocas variables, Baja = afecta muchas)')
    axes[1, 1].set_xlabel('Coeficiente de Variación')

    plt.tight_layout()
    plt.show()

    return global_importance, spec_df
```

```
def analyze_trajectory_sensitivity(trajectories_dict, X, feature_names):
    """
    Analiza sensibilidad en características de las trayectorias
    """
    print("\n" + "=" * 70)
    print("ANÁLISIS DE SENSIBILIDAD - CARACTERÍSTICAS DE TRAYECTORIAS")
    print("=" * 70)

    trajectory_results = {}

    for state_name, trajectories in trajectories_dict.items():
        print(f"\n🔍 Analizando trayectorias de: {state_name}")

        # Extraer características de las trayectorias
        trajectory_features = []

        for traj in trajectories:
            if len(traj) > 10: # Asegurar que hay suficientes puntos
                features = {
                    'valor_final': traj[-1],
                    'valor_maximo': np.max(traj),
                    'valor_minimo': np.min(traj),
                    'area_curva': np.trapz(traj),
                    'tasa_crecimiento_max': np.max(np.diff(traj)) if len(traj) > 1 else 0,
                    'tiempo_pico': np.argmax(traj) / len(traj) if len(traj) > 0 else 0
                }
                trajectory_features.append(list(features.values()))

        if len(trajectory_features) > 20: # Mínimo de muestras
            y_traj = np.array(trajectory_features)
            feature_names_traj = list(features.keys())
```

```
        # Analizar cada característica de trayectoria
        traj_importances = []
        for i, traj_feature_name in enumerate(feature_names_traj):
            y_feature = y_traj[:, i]

            if np.std(y_feature) > 1e-10: # Verificar variabilidad
                rf_traj = RandomForestRegressor(n_estimators=50, random_state=42)
                rf_traj.fit(X, y_feature)

                importance_df = pd.DataFrame({
                    'feature': feature_names,
                    'importance': rf_traj.feature_importances_
                }).sort_values('importance', ascending=False)

                traj_importances.append({
                    'feature_name': traj_feature_name,
                    'importance_df': importance_df
                })

        trajectory_results[state_name] = traj_importances

    # Mostrar resumen
    print(f" Características analizadas: {len(traj_importances)}")
    if traj_importances:
        top_overall = []
        for traj_imp in traj_importances:
            top_param = traj_imp['importance_df'].iloc[0]
            top_overall.append((top_param['feature'], top_param['importance']))
```

```
    # Parámetro más frecuente en top 1
    from collections import Counter
    top_params = [param for param, imp in top_overall]
    most_common = Counter(top_params).most_common(1)
    if most_common:
        print(f" Parámetro más influyente: {most_common[0][0]}")

    return trajectory_results
```

```
def generate_comprehensive_report(results, global_importance, spec_df, trajectory_results):
    """
    Genera un reporte completo con recomendaciones
    """
    print("=" * 80)
    print("📊 REPORTE COMPLETO DE SENSIBILIDAD")
    print("=" * 80)

    # 5.1 Parámetros más importantes globalmente
    print("\n🌐 PARÁMETROS MÁS IMPORTANTES GLOBALMENTE:")
    print(global_importance.head(8).to_string(index=False))

    # 5.2 Parámetros más específicos
    print("\n🔍 PARÁMETROS MÁS ESPECÍFICOS:")
    print(spec_df.head(5).to_string(index=False))

    # 5.3 Calidad predictiva por variable
    print("\n📈 CALIDAD PREDICTIVA POR VARIABLE:")
    r2_summary = []
    for state_name, result in results.items():
        r2_summary.append({
            'Variable': state_name,
            'R²': result['r2'],
            'MSE': f"{result['mse']:.2f}",
            'Top_Parametro': result['importance_df'].iloc[0]['feature']
        })

    r2_df = pd.DataFrame(r2_summary).sort_values('R²', ascending=False)
    print(r2_df.to_string(index=False))
```

```
def run_complete_multivariate_analysis(n_samples=200):
    """
    Ejecuta el análisis completo para todas las variables
    """
    print("🚀 INICIANDO ANÁLISIS MULTIVARIADO COMPLETO")
    print("=" * 70)

    # 1. Generar dataset completo
    print("📁 Generando dataset con todas las variables...")
    X, y_dict, feature_names, trajectories_dict = generate_complete_dataset(n_samples)

    # 2. Análisis de valores finales
    print("\n🔍 Realizando análisis de sensibilidad...")
    results = analyze_all_variables_sensitivity(X, y_dict, feature_names)

    # 3. Visualizaciones
    print("\n📊 Creando visualizaciones...")
    global_importance, spec_df = create_comparative_visualization(results, feature_names)

    # 4. Análisis de trayectorias (opcional - puede ser lento)
    print("\n📈 Analizando características de trayectorias...")
    trajectory_results = analyze_trajectory_sensitivity(trajectories_dict, X, feature_names)

    # 5. Reporte final
    generate_comprehensive_report(results, global_importance, spec_df, trajectory_results)

    return results, global_importance, trajectory_results
```

MATERIAL COMPLEMENTARIO

