



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA QUÍMICA, BIOTECNOLOGÍA Y
MATERIALES

**DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO PARA PREDICCIÓN
DE VÍAS METABÓLICAS ÓPTIMAS EN BASE A LA CARGA ENZIMÁTICA**

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN BIOTECNOLOGÍA

CATALINA BELÉN GUERRA CORNEJO

PROFESOR GUÍA:
ALVARO OLIVERA NAPPA

MIEMBROS DE LA COMISIÓN:
MARÍA PAZ CORTÉS BURGOS
JOSÉ SALGADO HERRERA
PEDRO SAA HIGUERA

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERA CIVIL EN
BIOTECNOLOGÍA
POR: **CATALINA BELÉN GUERRA CORNEJO**
FECHA: 2021
PROF. GUÍA: ALVARO OLIVERA NAPPA

DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO PARA PREDICCIÓN DE VÍAS METABÓLICAS ÓPTIMAS EN BASE A LA CARGA ENZIMÁTICA

El objetivo de este trabajo de título es diseñar e implementar un algoritmo que prediga vías metabólicas de producción minimizando la carga enzimática del organismo productor. Se define la carga enzimática como la cantidad de enzima necesaria para catalizar las reacciones de una ruta metabólica. Esta carga será representada a través de los parámetros cinéticos de las enzimas, pues la cinética tiene relación directa con la cantidad de enzima requerida.

Para llevar a cabo este trabajo se realizó una extracción de las actividades específicas reportadas en la base de datos **BRENDA**. Con esta información se planteó un problema de optimización donde se minimiza la carga enzimática utilizando los parámetros ya extraídos, luego, se compararon las distribuciones de flujos obtenidos para un modelo a escala genómica de *E. coli* con datos experimentales obtenidos de bibliografía. Finalmente, se acopló esta minimización a *optStoic*, un método de diseño de rutas metabólicas basado en estequiometría y minimización de flujos. Con el método implementado se evaluó su desempeño en tres casos de estudio: una red metabólica ficticia, evaluación de rutas del metabolismo central y la producción de compuestos isoprenoides con una base universal de reacciones.

Luego de plantear la carga enzimática como un problema de optimización con los datos extraídos, se obtiene una distribución de flujos similar a la reportada en literatura. Al incorporar esto a *optStoic* se diseñaron rutas para los escenarios mencionados. Los resultados en los dos primeros casos de estudio fueron los esperados al conocer de antemano las rutas a obtener, aunque se destaca la necesidad de especificar la estequiometría general en un comienzo. Para el tercer caso de estudio se aprecian dos limitaciones principales: la necesidad de conocer todos los compuestos de inicio y fin de la ruta con una estequiometría bien definida y la falta de actividad específica para más de la mitad de las reacciones presentes en la base universal. Esto conllevó a obtener rutas similares entre sí, éstas no se alejan mucho de la ruta original en la cual se basó la estequiometría impuesta y con algunas reacciones sin carga enzimática asociada. Al ampliar la estequiometría aparecen reacciones adicionales que no aportan ni generan diversidad en las rutas resultantes, por lo que esta no fue una solución para aportar mayor diversidad a las soluciones.

La herramienta desarrollada en este trabajo se encuentra funcional y se logró estudiar alternativas al caso de producción de isoprenoides pero presenta diversas limitaciones, cumpliendo parcialmente el objetivo propuesto en su estado actual y se postulan diversos trabajos a futuro para mejorar su desempeño.

Agradecimientos

Quiero agradecer a mi familia. A mi madre, María Teresa, que siempre ha sido una inspiración para mí, como mujer, como persona y como profesional. A mi padre, Patricio, que siempre me ha enseñado con su gran ejemplo a esforzarme por lo que quiero para mi vida, no importa lo duro que sea el camino, el trabajo todo lo vence. A mi hermano, Diego, que sin hablar nos entendemos y reímos de nuestros chistes internos, pero también sabemos cuando el otro necesita apoyo. Ustedes tres han sido mi núcleo y son parte de todo lo que he emprendido. Una no escoge a la familia, pero la familia tampoco te escoge a tí, espero se sientan tan afortunados como yo me siento de tenerlos a ustedes. Quiero agradecer a Damián por su apoyo incondicional todos estos años de universidad y en este último año que no ha sido fácil, por ayudarme a pensar con calma cuando sentía que tenía el mundo sobre mis hombros y por decirme que soy seca tantas veces que terminé creyéndolo. Quiero agradecer también a mi compañera perruna, Aina. Llegaste el mismo año que entré a la facultad y me acompañaste en todos mis ratos de estudio. Quiero agradecer también a mis profesores del colegio, Emilio, Ricardo, Pedro y Sandy. Les agradezco por ver en mí el potencial que creyeron que tenía, me incentivaron a tomar desafíos nuevos e ir siempre un poco más allá. Los guardo con mucho cariño en mi corazón.

Quiero agradecer a toda mi familia de Taekwondo Místico. A mi maestro Bastián, que me conoció en octavo básico y hasta el día de hoy sigo aprendiendo de él. A los maestros Adolfo y Darío, que también me han ayudado a ser mejor cada día. A mis compañeros y amigos Javier, Sthepy, Karina, Isi y Fer, por tantos entrenamientos, campamentos, torneos y buenos momentos que hemos compartido estos años. También quiero agradecer a todos los que han sido mis alumnos y alumnas estos años, empecé con mi escuela estando en mi primer año de especialidad y los he visto crecer tanto como ustedes me han visto crecer a mí. Cada vez que estuve sin ánimo de nada por un mal día, con sus sonrisas me llenaron de energía para hacer clases y ayudarlos a mejorar. Enseñando he aprendido mucho de ustedes. Quiero agradecer a todos los amigos que hice durante mi paso por la facultad. A mi team querido: Naro, Gino y Pali. Por tantas tardes y noches de estudio, los mensajes a las 6:00 AM de 'ya no llegué a las 8:00' porque todos vivimos lejos, tantas risas durante los tecitos para sacar la vuelta, por los colapsos y rabias con tanto estrés a final de semestre, por los carretes de los viernes y los viajes a la playa, por las salidas a comer pastelitos del Kanela o las papas de los egipcios que siempre compartíamos 'fiándonos' entre nosotros. A mis amigos del DIQBM Amanda, Denisse, Nico, Benja A., Benja P., Gera, Basti, Consu, Javi, Seba y Kevin. Ojalá nos encontremos en el futuro como profesionales.

Finalmente, agradecer al Centro de Modelamiento Matemático (CMM) por el financiamiento recibido del fondo AFB-17001, ANID, Chile.

Tabla de Contenido

1. Introducción	1
2. Antecedentes generales	3
2.1. Planteamiento general de un modelo metabólico	3
2.1.1. Análisis de balance de flujos	3
2.1.2. Modelos metabólicos a escala genómica	5
2.2. Compuestos isoprenoides	6
2.3. Revisión bibliográfica	9
2.3.1. Análisis de redes metabólicas	9
2.3.2. Métodos de estudio de rutas metabólicas	11
3. Objetivos	16
4. Metodología	17
4.1. Obtención de parámetros cinéticos	17
4.2. Modelamiento de la carga enzimática	18
4.3. Elección del método base para el desarrollo del algoritmo e incorporación de la carga enzimática	19
5. Resultados	23
5.1. Evaluación de la carga enzimática en un <i>GSM</i>	23
5.2. Desempeño del algoritmo	28
5.2.1. Prueba de concepto	29
5.2.2. Metabolismo central	32
5.2.2.1. Producción de gliceraldehído 3-fosfato a partir de glucosa	32
5.2.2.2. Producción de piruvato a partir de distintas fuentes	35
5.2.3. Base universal de reacciones: producción de isopentenil difosfato (<i>IPP</i>)	41
5.2.3.1. Producción de <i>IPP</i> minimizando flujos	42
5.2.3.2. Producción de <i>IPP</i> minimizando carga enzimática	47
5.3. Discusiones generales	53
6. Conclusiones	56
7. Nomenclatura	58
Bibliografía	59
8. Anexos	62

8.1.	Obtención de parámetros (BRENDA)	62
8.2.	Modelamiento de la carga enzimática	69
8.3.	Simulación de la carga enzimática en un <i>GSM</i> y comparación con datos experimentales	71
8.4.	Modificaciones a <i>optStoic</i>	76
8.5.	Reacciones consideradas en el metabolismo central	85
8.6.	Casos de estudio adicionales	88
8.6.1.	Producción de piruvato a partir de glucosa	88
8.6.2.	Producción de piruvato a partir de distintas fuentes (límites holgados)	91
8.6.3.	Pruebas del algoritmo para producir <i>IPP</i> a partir de Acetil-CoA con límites holgados	95
8.6.4.	Rutas adicionales para la producción de <i>IPP</i> minimizando flujos . . .	97
8.6.5.	Producción de mevalonato a partir de acetil-CoA	100
8.6.5.1.	Producción de mevalonato minimizando flujos	100
8.6.5.2.	Producción de mevalonato minimizando carga enzimática	105

Índice de Tablas

5.1.	Resumen de los parámetros cinéticos filtrados	23
5.2.	Resumen de los ΔG_{rxn} y actividades específicas inversas.	24
5.3.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida, prueba de concepto	29
5.4.	Cargas enzimáticas para producir F a partir de A	31
5.5.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de gliceraldehído 3-fosfato a partir de glucosa	33
5.6.	Cargas enzimáticas para la producción de gliceraldehído a partir de glucosa . .	34
5.7.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de piruvato a partir de distintas fuentes de carbono	35
5.8.	Cargas enzimáticas para la producción de piruvato a partir de distintas fuentes, límites estrictos	39
5.9.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de <i>IPP</i> a partir de Acetil-CoA	41
5.10.	Cargas enzimáticas para la producción de <i>IPP</i> a partir de acetil CoA (Ruta 1)	47
5.11.	Cargas enzimáticas para la producción de <i>IPP</i> a partir de acetil CoA (Ruta 2)	48
5.12.	Cargas enzimáticas para la producción de <i>IPP</i> a partir de acetil CoA (Ruta 3)	49
5.13.	Cargas enzimáticas para la producción de <i>IPP</i> a partir de acetil CoA (Ruta 4)	50
5.14.	Cargas enzimáticas para la producción de <i>IPP</i> a partir de acetil CoA (Ruta 5)	51
8.1.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b4025 (proteína PGI)	71
8.2.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b2097 (proteína fbaB)	71
8.3.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b1723 (proteína pfkB)	72
8.4.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b1854 (proteína pykA)	72
8.5.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b1702 (proteína ppsA)	72
8.6.	Reacciones presentes en red metabólica del metabolismo central	87
8.7.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de piruvato a partir de glucosa	88
8.8.	Cargas enzimáticas para la producción de piruvato a partir de glucosa	89
8.9.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de piruvato a partir de distintas fuentes de carbono	91
8.10.	Cargas enzimáticas para la producción de piruvato a partir de distintas fuentes, límites holgados.	93

8.11.	Rangos de tasas de producción o consumo para los compuestos de entrada y salida para la producción de Mevalonato a partir de Acetil-CoA	100
8.12.	Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 1)	105
8.13.	Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 2)	106
8.14.	Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 3)	107
8.15.	Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 4)	107
8.16.	Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 5)	108

Índice de Ilustraciones

2.1.	Construcción de la matriz estequiométrica.	3
2.2.	Representación gráfica de un <i>FBA</i>	4
2.3.	Representación gráfica de la red metabólica de un <i>GSM</i> (<i>E. coli core</i>).	6
2.4.	Compuestos isoprenoides: unidad básica y ejemplos de monoterpenos.	7
2.5.	Biosíntesis de IPP mediante (A) ruta del mevalonato y (B) ruta <i>DXP</i>	8
2.6.	Representación gráfica del método <i>optStoic</i>	14
2.7.	Ecuaciones principales del método <i>ECM</i>	15
5.1.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b4025 (proteína PGI)	25
5.2.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b2097 (proteína fbaB)	26
5.3.	Resultados prueba de concepto: producción de F a partir de A	30
5.4.	Resultados del algoritmo en prueba de producción de gliceraldehído 3-fosfato	33
5.5.	Resultados del algoritmo en prueba de producción de 2 moles de piruvato minimizando flujos	36
5.6.	Resultados del algoritmo en prueba de producción de 2 moles de piruvato minimizando carga enzimática	38
5.7.	Ruta 1: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos. La ruta 2 es análoga, utilizando <i>NADPH</i> como transportador de electrones (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).	42
5.8.	Ruta 3: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos. La ruta 4 es análoga, utilizando <i>NADPH</i> como transportador de electrones (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).	43
5.9.	Ruta 5: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos. La ruta 7 es análoga, utilizando <i>NADH</i> como transportador de electrones (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	43
5.10.	Ruta 6: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos. La ruta 10 es análoga, utilizando <i>NADH</i> como transportador de electrones (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	44
5.11.	Ruta 8: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos. La ruta 9 es análoga, utilizando <i>NADPH</i> como transportador de electrones (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	45
5.12.	Ruta 1: producción de <i>IPP</i> a partir de Acetil-coA minimizando carga enzimática (flujo total: $10 \frac{mmol}{gr S_0 \cdot hr}$).	47
5.13.	Ruta 2: producción de <i>IPP</i> a partir de Acetil-coA minimizando carga enzimática (flujo total: $11 \frac{mmol}{gr S_0 \cdot hr}$).	48
5.14.	Ruta 3: producción de <i>IPP</i> a partir de Acetil-coA minimizando carga enzimática (flujo total: $11 \frac{mmol}{gr S_0 \cdot hr}$).	49

5.15.	Ruta 4: producción de <i>IPP</i> a partir de Acetil-coA minimizando carga enzimática (flujo total: $10 \frac{mmol}{gr S_0 \cdot hr}$).	50
5.16.	Ruta 5: producción de <i>IPP</i> a partir de Acetil-coA minimizando carga enzimática (flujo total: $9 \frac{mmol}{gr S_0 \cdot hr}$).	51
8.1.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b1723 (proteína pfkB)	73
8.2.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b1854 (proteína pykA)	74
8.3.	Comparación del modelo con datos experimentales de bibliografía: <i>knock out</i> gen b1702 (proteína ppsA)	75
8.4.	Rutas para la producción de piruvato a partir de glucosa, vía glicólisis	89
8.5.	Resultados del algoritmo en prueba de producción de 1-2 moles de piruvato minimizando flujos	92
8.6.	Resultados del algoritmo en prueba de producción de 1-2 moles de piruvato minimizando carga enzimática	93
8.7.	Producción de isopentenil difosfato a partir de acetil-CoA minimizando flujos (Ruta 1)	95
8.8.	Producción de isopentenil difosfato a partir de acetil-CoA minimizando flujos (Ruta 2)	96
8.9.	Ruta 2: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).	97
8.10.	Ruta 4: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).	97
8.11.	Ruta 7: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	98
8.12.	Ruta 9: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	98
8.13.	Ruta 10: producción de <i>IPP</i> a partir de Acetil-coA minimizando flujos (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	99
8.14.	Ruta 1: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $3 \frac{mmol}{gr S_0 \cdot hr}$).	100
8.15.	Ruta 2: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $3 \frac{mmol}{gr S_0 \cdot hr}$).	101
8.16.	Ruta 3: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).	101
8.17.	Ruta 4: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).	101
8.18.	Ruta 5: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).	102
8.19.	Ruta 6: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).	102
8.20.	Ruta 7: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).	103
8.21.	Ruta 8: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).	103
8.22.	Ruta 9: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).	103

8.23.	Ruta 10: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).	104
8.24.	Ruta 1: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	105
8.25.	Ruta 2: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $9 \frac{mmol}{gr S_0 \cdot hr}$).	106
8.26.	Ruta 3: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).	106
8.27.	Ruta 4: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).	107
8.28.	Ruta 5: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $8 \frac{mmol}{gr S_0 \cdot hr}$).	108

Capítulo 1

Introducción

La biología de sistemas estudia y analiza los sistemas biológicos en toda su complejidad, considerando interacciones entre los distintos componentes de un sistema biológico, mecanismos de regulación y señalización. Por otro lado, la ingeniería metabólica se enfoca en el diseño de sistemas biológicos orientados a un objetivo en particular como la producción de un compuesto deseado. Ambas áreas de investigación plantean, analizan y estudian los procesos biológicos de forma matemática y computacional, considerando interacciones entre los distintos componentes de la célula y con el medio que la rodea externamente. Debido a la complejidad de los sistemas biológicos, el planteamiento de modelos para el estudio y comprensión de dichos procesos se vuelve fundamental [1].

Por otro lado, con el desarrollo de nuevas tecnologías ha sido posible realizar estudios experimentales para obtener información a nivel de genoma, transcriptoma, metaboloma y proteoma, con la cual se han desarrollado métodos para su análisis [1]. Su integración a los modelos de sistemas biológicos se vuelve aún más atractiva para representar fielmente organismos de interés y poder predecir sus comportamientos ante distintas condiciones.

Una de las aplicaciones de estas tecnologías es el uso de herramientas computacionales para guiar el diseño de nuevas rutas de producción, las cuales deben tener en consideración la factibilidad termodinámica, la disponibilidad de enzimas para dichos fines, el rendimiento, el gasto energético, entre otros factores [2]. Hasta la fecha, se han planteado distintos modelos de organismos que son de uso común en la producción de compuestos de alto interés comercial e industrial [3, 4] y se han postulado distintas metodologías para diseñar rutas metabólicas *de novo* [2].

Dentro de las distintas herramientas mencionadas, se encuentran los modelos metabólicos. Éstos buscan representar el metabolismo de una célula incorporando información sobre sus reacciones, enzimas involucradas y genes asociados [3]. Cuando un modelo es pensado para un organismo particular, permite entender cómo se enfrenta a distintas condiciones ambientales. Los modelos pueden ser combinados con distintas estrategias de diseño de rutas metabólicas, aprovechando su especificidad. Las metodologías de diseño se suelen enfocar en un objetivo, por ejemplo, maximizar la producción de biomasa o algún compuesto en particular [5], observar el efecto de ciertas deleciones de genes [6] o la factibilidad termodinámica de una ruta metabólica [7].

En este trabajo se busca generar un algoritmo que prediga vías metabólicas de producción minimizando la carga enzimática, entendiendo por carga enzimática la cantidad de enzima requerida para procesar el sustrato correspondiente. Luego de su implementación se pondrá a prueba en tres conjuntos de reacciones distintos: un conjunto genérico con reacciones ficticias para corroborar el funcionamiento del algoritmo, otro conjunto que involucre solo las reacciones del metabolismo central para evaluar rutas metabólicas ya conocidas; y un tercer conjunto que considere un universo de reacciones que contemple todas las reacciones documentadas en las bases de datos biológicas disponibles y enfrentar un desafío real de la industria de producción recombinante. Para este último conjunto de reacciones, se considerará la producción de compuestos isoprenoides y/o sus precursores a partir glucosa para diseñar nuevas rutas metabólicas que faciliten su producción en microorganismos.

Capítulo 2

Antecedentes generales

2.1. Planteamiento general de un modelo metabólico

En ingeniería metabólica se busca producir compuestos y materias primas de alto interés comercial e industrial aprovechando los mecanismos de producción biológica que ofrecen las células, ya sea con los mecanismos propios del organismo o con la inserción de nuevos mecanismos [8]. El desarrollo de distintas metodologías y herramientas computacionales ha permitido mejorar la predicción y el diseño de estas nuevas rutas de producción [2].

Una representación típica del metabolismo de una célula corresponde a la matriz estequiométrica S . En la posición (i,j) de esta matriz está el coeficiente estequiométrico del metabolito i en la reacción j , los reactivos de las reacciones tienen signo negativo, mientras que los productos tienen signo positivo. Multiplicando esta matriz por un vector de flujos para cada reacción se obtienen los balances de masa de cada compuesto involucrado en la red metabólica. En la Figura 2.1 se explica gráficamente cómo se construye la matriz S :

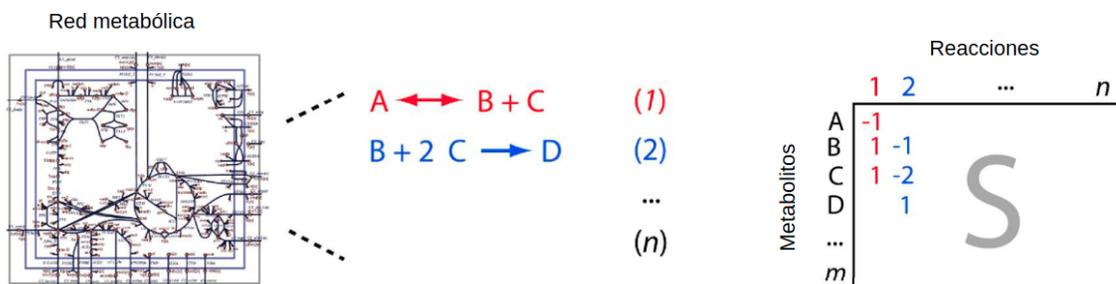


Figura 2.1: Construcción de la matriz estequiométrica, adaptado de Orth et al. (2010)

2.1.1. Análisis de balance de flujos

Para analizar una red metabólica se utiliza el Análisis de Balance de Flujos o *FBA* (*Flux Balance Analysis*) [5]. Este es un problema de optimización que se plantea de la siguiente forma:

$$\max \quad Z = c^T \cdot v \quad (2.1)$$

$$\text{s.a.} \quad S \cdot v = 0 \quad (2.2)$$

$$LB_j \leq v_j \leq UB_j \quad , \forall j = 1, \dots, n \quad (2.3)$$

Donde:

- v es un vector que contiene los flujos de todas las reacciones presentes en la red metabólica.
- Z es la función objetivo, compuesta por el vector de flujos v y un ponderador de pesos c^T para cada componente de dicho vector. En general, se maximiza una reacción en particular, que puede ser la producción de un compuesto en particular.
- S es la matriz estequiométrica.
- LB_j y UB_j son los límites inferior y superior (*Lower Bound* y *Upper Bound*), respectivamente, que acotan el valor del flujo para cada reacción j , pueden estar dados por datos experimentales, disponibilidad de sustrato o un rango arbitrario dependiendo de la información disponible.

Al resolver el problema se obtiene una distribución de flujos de la red metabólica para la cual se optimiza la función objetivo escogida (Ec. 2.1), manteniendo estado estacionario para todos los metabolitos de la red (Ec. 2.2) y los flujos dentro de un rango definido para cada reacción (Ec. 2.3). El estado estacionario se mantiene al igualar la Ecuación 2.2 a cero, cada compuesto se produce y se consume a la misma tasa sin generar acumulación. En la Figura 2.2 se representa gráficamente en un ejemplo simplificado de cómo se genera el espacio factible de soluciones para el problema de optimización, utilizando las restricciones escogidas para cada componente del vector de flujos. Dentro de este conjunto se obtiene la o las soluciones que optimizan la función objetivo escogida.

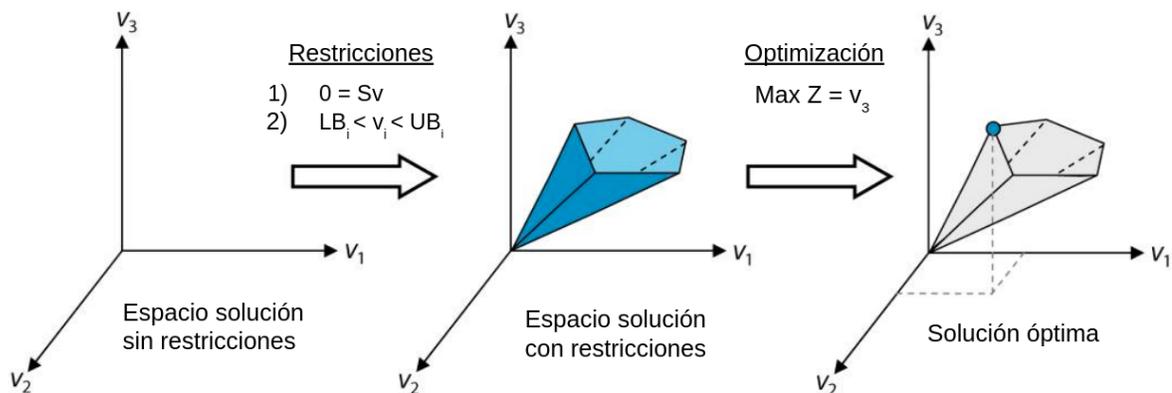


Figura 2.2: Representación gráfica de un *FBA*, adaptado de Orth et al. (2010)

2.1.2. Modelos metabólicos a escala genómica

Con el fin de representar un organismo en toda su complejidad, se plantean los modelos a escala genómica (*GSM*, por sus siglas en inglés). La construcción de estos modelos involucra toda la información a nivel del genoma disponible y de las reacciones conocidas del organismo en cuestión, con ello se reconstruye la red metabólica que lo representa. Dado que no existe una reacción explícita para la formación de biomasa, se genera una reacción adicional que la represente basándose en la composición celular del organismo de interés. Si además se dispone de otra información ómica disponible, como el transcriptoma o proteoma, también es incluido en el *GSM*. Debido a que la disponibilidad de información varía de un organismo a otro, estos modelos pueden tener distintos alcances o limitaciones. Si, por ejemplo, no se tiene la composición celular del organismo en estudio, se realizan aproximaciones [3].

En la Figura 2.3 se observa un ejemplo de un modelo a escala genómica de *Escherichia coli* (*E. coli core*) que incluye todo el metabolismo central [9]. Cada línea azul corresponde a una reacción y cada punto a un metabolito en particular. Cada una de estas reacciones es catalizada por una enzima que, a su vez, está asociada a uno o más genes. Es a partir de estas asociaciones gen-proteína-reacción (*GPR*) que se logra inferir qué reacciones están presentes en un organismo en particular.

Una vez que el modelo es construido y validado con información experimental, se utiliza para predecir y analizar los flujos de las vías metabólicas presentes, detectar genes esenciales para el organismo, diseñar nuevas vías de producción, estudiar rutas alternativas para precursores de interés, predecir el comportamiento del microorganismo ante distintos agentes de estrés medioambiental, entre otras aplicaciones [4].

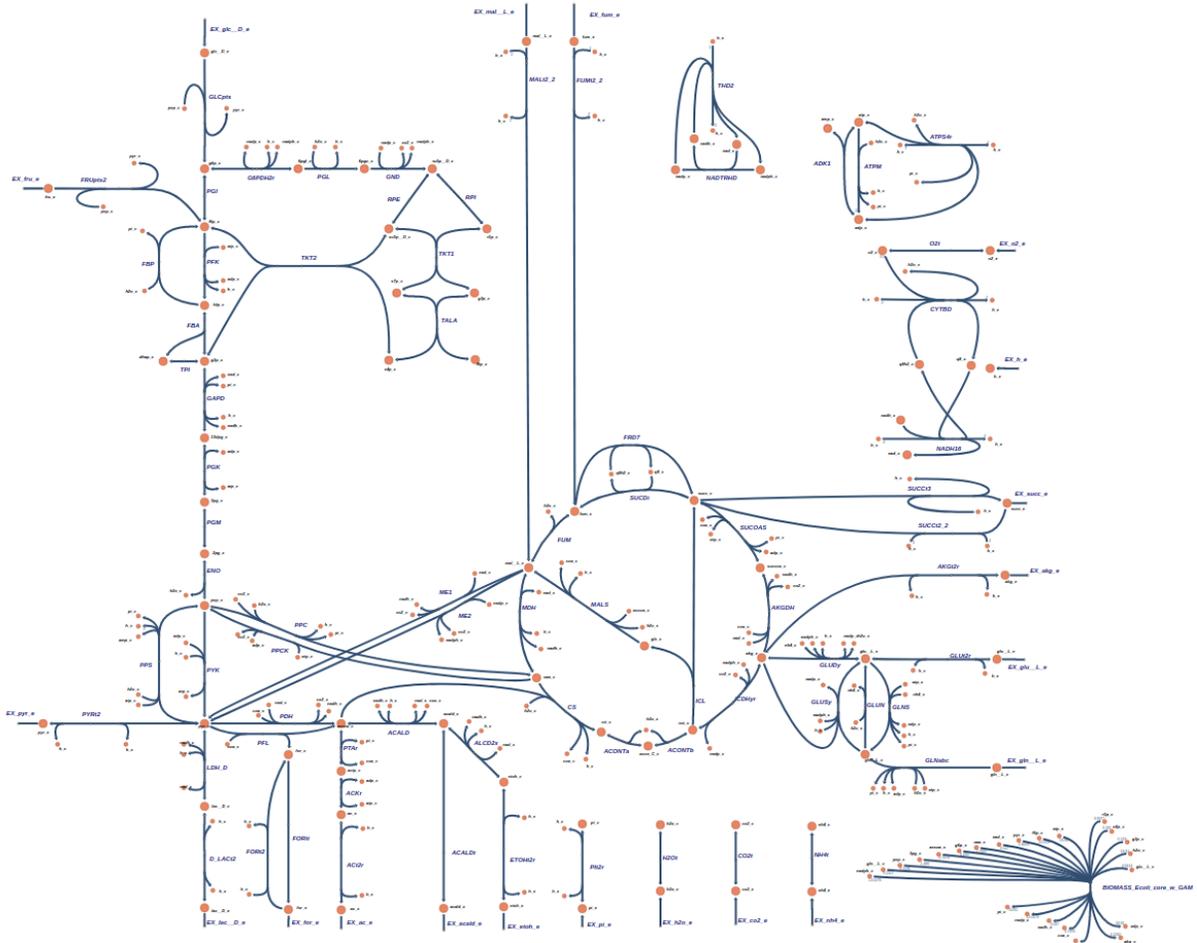
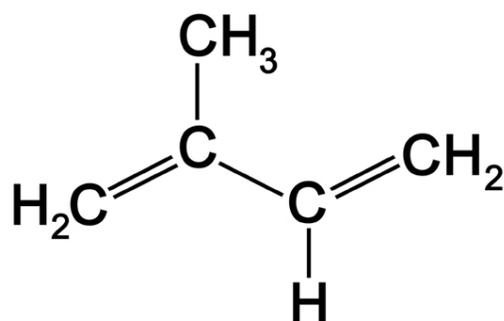


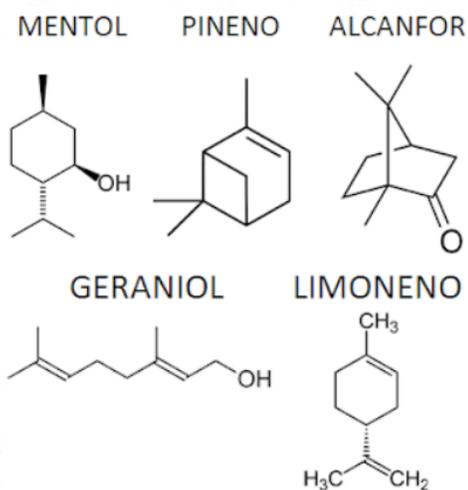
Figura 2.3: Representación gráfica de la red metabólica de un *GSM*, obtenida de BiGG Models

2.2. Compuestos isoprenoides

Un ejemplo de compuestos a producir utilizando ingeniería metabólica son los llamados compuestos isoprenoides, o también llamados terpenos. Estos son polímeros que tienen como unidad básica el isopreno, un compuesto orgánico formado por carbono e hidrógeno (Figura 2.4.a). Dependiendo de la cantidad de monómeros presentes en la estructura, estos pueden formar compuestos volátiles, vitaminas liposolubles, pigmentos carotenoides, entre otros [10]. En la Figura 2.4.b se aprecian ejemplos de monoterpénos, es decir, compuestos conformados por la unión de dos isoprenos, todos ellos constituyen parte de los olores característicos en distintas plantas y frutos.



(a) Estructura química del isopreno



(b) Distintos tipos de monoterpenos: mentol, pineno, alcanfor, geraniol, limoneno

Figura 2.4: Compuestos isoprenoides: unidad básica y ejemplos de monoterpenos.

La producción de los distintos tipos de compuestos isoprenoides es de alto interés comercial, sobretodo para las industrias alimentarias y de perfumería dadas las características organolépticas que aportan [11, 12]. Debido a que su producción de forma natural se realiza principalmente en plantas, el uso de microorganismos para la producción de forma recombinante de estos compuestos es cada vez más atractivo para lograr mayores rendimientos [12].

Uno de los principales precursores es el isopentenil difosfato (*IPP*, por sus siglas en inglés), el cual puede ser sintetizado por dos vías distintas: a partir de acetil-CoA por la ruta del mevalonato o bien, a partir de piruvato y gliceraldehído 3-fosfato por la ruta de la deoxixilulosa 5-fosfato (*DXP*, por sus siglas en inglés) [13, 14]. Estas rutas se pueden apreciar en la Figura 2.5. Los precursores de ambas rutas son puntos de control importantes en el metabolismo central. De ambas opciones, la ruta preferida para realizar producción recombinante es la del mevalonato, pues la ruta *DXP* tiene genes asociados que solo están presentes en células con plastidios y utiliza proteínas como reactivos en ciertas reacciones de la ruta [14]. Esto hace más difícil lograr su producción utilizando los microorganismos más típicos en la producción recombinante como *Escherichia coli* o *Saccharomyces cerevisiae* [15].

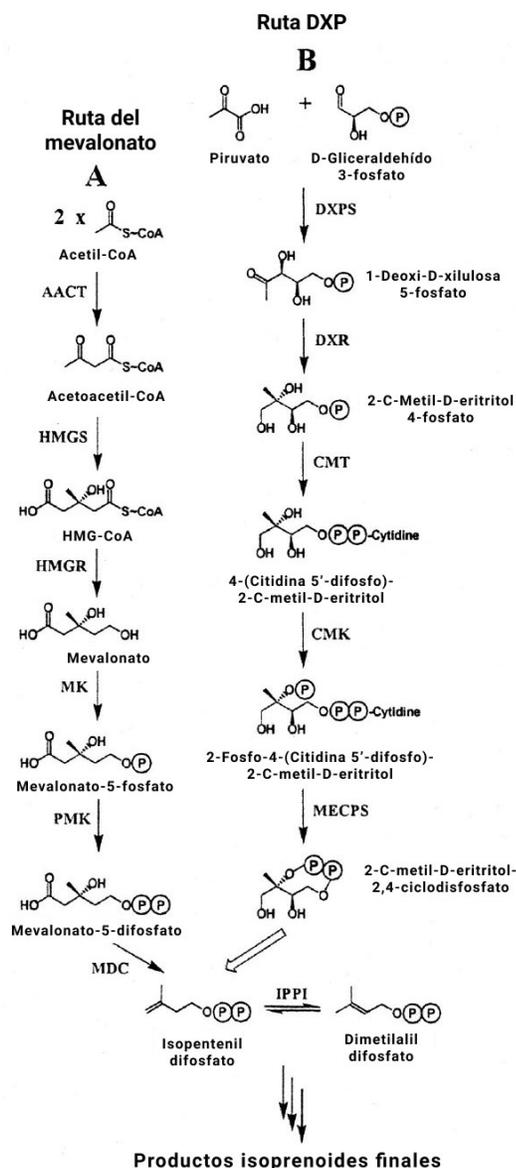


Figura 2.5: Biosíntesis de IPP mediante (A) ruta del mevalonato y (B) ruta *DXP*, adaptado de Lange et al. (2000)

Dentro la ruta del mevalonato (Figura 2.5(A)), los pasos previos a la formación de este compuesto constituyen el principal foco al intentar aumentar la producción de *IPP*. La formación de acetoacetil-CoA constituye un paso que termodinámicamente es desfavorable, por lo que se han buscado caminos alternativos [16]. Por otro lado, la formación de mevalonato a partir de 3-hidroxi-3-metilglutaril-CoA (*HMG-CoA*, por sus siglas en inglés) es catalizada por la enzima *HMG-CoA* reductasa, la cual está sometida a mecanismos de regulación que redireccionan el flujo hacia otras rutas metabólicas y disminuyen la producción de *IPP* [12, 17].

Ya existen sistemas de producción de compuestos isoprenoides en *Escherichia coli* [18], pero también se busca lograr su producción en otros microorganismos que logren generar

compuestos isoprenoides más complejos por tener otros mecanismos de producción [12]. El mayor desafío es que muchos de estos potenciales organismos hospederos presentan compartimentos celulares que pueden concentrar en su interior ciertos metabolitos. En este caso, siendo el acetil-CoA el precursor principal, al utilizar *E. coli* como hospedero no existe este problema, pero en otros organismos como *Saccharomyces cerevisiae* se concentra dentro de las mitocondrias y no se exporta al citosol. Además, por los distintos mecanismos de regulación asociados a las enzimas que catalizan la ruta del mevalonato, es necesaria la acumulación de los precursores para lograr mayores producciones de *IPP* [17].

2.3. Revisión bibliográfica

Al plantear un modelo metabólico se toman en cuenta las reacciones que puede llevar a cabo el organismo que se desea simular. Además del metabolismo central, pueden existir reacciones catalizadas por enzimas y reguladas por genes que son propios de dicho organismo y que deben estar presentes para representar de forma plausible a la especie en estudio. Una vez que se tiene el conjunto de reacciones que conformarán la red metabólica, existen distintos análisis que se pueden realizar para simular una situación deseada y así predecir el comportamiento de una célula ante un escenario dado [19, 20, 21, 22, 23, 24].

Por otra parte, se han hecho esfuerzos por generar herramientas que sirvan para diseñar rutas metabólicas de producción tomando en cuenta distintos factores como la presencia o ausencia de las reacciones en el organismo a utilizar, la factibilidad termodinámica, la estequiometría de las reacciones, entre otros [7, 25, 26, 27].

En esta sección se llevará a cabo una revisión de ambos tópicos mencionados, explicando brevemente el funcionamiento de distintas herramientas y en qué se basaron al momento de ser desarrolladas.

2.3.1. Análisis de redes metabólicas

Un *FBA* es una distribución de flujos para la cual se optimiza la función objetivo escogida, este resultado está sujeto a distintos supuestos. El primero es el de estado estacionario para los metabolitos intermedios [5]. Este supuesto implica que los compuestos que están presentes en distintas reacciones de la red pero no cuentan con una reacción de intercambio con el medio externo, ya sea de entrada o salida, no se acumulan dentro del organismo. El segundo es la existencia de un objetivo único a cumplir [5], esto no necesariamente ocurre en la célula, sobretodo en condiciones adversas o de falta de nutrientes donde el crecimiento celular y sus funciones se compensan según la disponibilidad de recursos y otros factores de estrés [28, 29, 30]. Otro supuesto fuerte es la existencia de un límite inferior y superior conocido que debe asignarse al flujo de cada reacción [5]. Al ser un problema de optimización cada componente puede tomar por separado un valor extremo, sin embargo, estos límites suelen ser holgados ya que generalmente no se tiene un rango medido experimentalmente para fijar dichos límites. La existencia de estos límites tiene un sentido biológico al restringir la capacidad máxima de las enzimas para catalizar una reacción, pero los valores utilizados para establecer estos límites no necesariamente están basados en esta capacidad.

Es importante recalcar que para la resolución de este problema, los límites impuestos a los flujos son los que definen el conjunto factible del problema de optimización. Por la dificultad de medir dichos flujos experimentalmente, es un problema desafiante el escoger apropiadamente estas cotas para la obtención de una distribución de flujos que sea plausible. Existen métodos de cuantificación de estos flujos intracelulares [31], donde se ha logrado obtener una distribución general de la red metabólica en distintos escenarios de delección de genes para entender mejor cómo influyen estas perturbaciones en los flujos observados [32]. Sin embargo, dichos experimentos requieren una gran cantidad de recursos, personal e insumos y otorgan resultados para el microorganismo y el escenario particular estudiado, por lo que no son datos útiles en otros casos de estudio.

Existen métodos que buscan aumentar la verosimilitud del *FBA*, ejemplos de esto son los métodos *Flux variability Analysis (FVA)* [19] *Parsimonious FBA (pFBA)* [20], *Minimization of metabolic adjustment (MOMA)* [19], *Regulatory On/Off Minimization (ROOM)* [21] y *Dynamic FBA (dFBA)* [22].

Dado que puede existir más de una distribución de flujos para lograr un cierto valor objetivo, *FVA* sigue la misma lógica que un *FBA* pero encuentra los rangos para el flujo de cada reacción de la red que permiten alcanzar el mismo valor objetivo en lugar de obtener una única distribución, entregando más información sobre qué reacciones tienen más o menos variabilidad en relación a cierto óptimo [19].

pFBA agrega al problema de optimización original (*FBA*) la minimización de la suma de los flujos en el objetivo. Este enfoque se basa en la idea de que flujos altos tienen un costo mayor al necesitar mayor cantidad de enzimas que catalicen las reacciones, por lo que la célula trata de minimizar la distribución de flujos general y a la vez maximizar el objetivo original [20].

Siguiendo con la misma idea de las distintas distribuciones de flujos que llegan al óptimo, el saber cuál es el efecto inmediato de deleccionar un gen es un problema de interés, ya que actúan distintos cambios en la regulación del organismo que buscan volver al óptimo redistribuyendo sus flujos de otra forma. El método *MOMA* predice la distribución de flujos inmediatamente después de la perturbación o *knock out*, esto lo hace minimizando la distancia entre la distribución de flujos inicial y la distribución con la perturbación [19].

El método *ROOM* entrega una distribución de flujos en estado estacionario luego de uno o más *knock outs*. Utiliza una distribución de referencia, por ejemplo, el resultado de un *FBA* para la cepa silvestre, y busca minimizar la cantidad de reacciones activas con respecto a la referencia entregada [21]. Generalmente se utiliza para evaluar el posible impacto que tiene el *knock out* de uno o más genes en un modelo metabólico y entender su relevancia para el funcionamiento del organismo.

Finalmente, *dFBA* busca realizar el mismo estudio de un *FBA* manteniendo el estado pseudo-estacionario para el medio intracelular y agregando modelos dinámicos para el medio extracelular. Estos modelos dinámicos suelen ser en base a cinética o datos experimentales de cultivos similares al simulado [22].

Todos estos métodos complejizan el planteamiento inicial del *FBA* y permiten abordar escenarios más específicos, pero siguen teniendo los mismos supuestos sobre los límites en el vector de flujos y sus resultados solo pueden asociarse al organismo estudiado.

En los *GSM* se pueden variar distintos parámetros para simular distintas condiciones ambientales, escenarios de delección de genes y aplicar lo análisis mostrados anteriormente, pero solo responden al organismo en el cual se basó su desarrollo, limitando la capacidad de generalizar sus resultados. Esto es un desafío a la hora de diseñar y predecir rutas de producción de compuestos que son de alto interés comercial e industrial, ya que pueden requerir reacciones o genes heterólogos para su producción. Además, *FBA* no considera información cinética de las reacciones para su planteamiento, por lo que se pierde esta información en el uso de estos modelos basados en restricciones. Con *dFBA* si se considera la cinética pero solo del medio externo.

2.3.2. Métodos de estudio de rutas metabólicas

Buscando diseñar rutas metabólicas para producir un compuesto de interés en un hospedero puntual, se planteó el método *OptStrain* que busca diseñar sistemas de producción microbianos a partir de un modelo a escala genómica, un conjunto de reacciones posibles y un metabolito a producir [25]. Este planteamiento requiere una separación manual del universo de reacciones entre nativas y no nativas que va a depender de cada organismo. Consta de 4 pasos principales:

- Se extrae la información de todas las reacciones publicadas en una base de datos universal, por ejemplo, **KEGG** [33].
- Se establece un conjunto de sustratos posibles y se plantea un Problema de Programación Lineal (*LP*, por sus siglas en inglés) donde se busca maximizar la producción de un compuesto de interés. Se consideran restricciones similares a un *FBA* y una restricción donde se normalizan los flujos obtenidos por una unidad de masa.
- Con este valor óptimo del producto celular deseado, se plantea un Problema de Programación Lineal-Entera (*MILP*, por sus siglas en inglés) que busca minimizar la cantidad de reacciones externas al microorganismo, donde las reacciones de la base de datos generada son separadas entre nativas y no nativas.
- Finalmente, se completa el trabajo utilizando otra metodología llamada *optKnock* que busca maximizar el flujo utilizado para producir el compuesto de interés eliminando ciertos genes asociados a reacciones que disminuyen este flujo objetivo [6].

Por otro lado, se plantea el método de *Max-min Driving Force (MDF)* para evaluar la factibilidad termodinámica de una ruta metabólica en específico sin la necesidad de contar con la información de la cinética de cada reacción involucrada [23]. Este método en particular busca solo evaluar la factibilidad de la ruta especificada. Como no es capaz de diseñar rutas *de novo*, se utiliza de forma complementaria a otros métodos de diseño. La energía de Gibbs disipada en una reacción ΔG_r se relaciona con el flujo de la misma según la ecuación 2.4.

$$\Delta G_r = -R \cdot T \cdot \ln \left(\frac{J^+}{J^-} \right) \quad (2.4)$$

Donde J^+ y J^- son los flujos de cada reacción en el sentido directo e inverso, respectivamente. Según esta relación, una enzima que cataliza una reacción muy lejos del equilibrio ($\Delta G_r \ll 0$) el flujo en el sentido inverso de la reacción es despreciable, mientras que si se encuentra cerca del equilibrio ($\Delta G_r \sim 0$) se pierde una cantidad considerable de enzima que está realizando la reacción en el sentido inverso. En base a esto, se entiende $-\Delta G_r$ como la fuerza impulsora de una reacción o *Driving Force*. Pasando a notación matricial, para poder evaluar todas las reacciones de una ruta, el cálculo de ΔG_r queda expresado en la ecuación 2.5:

$$-\Delta G_r = -(G^\circ + R \cdot T \cdot S^T x) \quad (2.5)$$

Donde G° es el vector de energía de Gibbs estándar, S^T es la matriz estequiométrica traspuesta y x es el vector de concentraciones logarítmicas, es decir, la componente x_j corresponde al logaritmo natural de la concentración del compuesto j . Vale decir que los valores que puede tomar el vector x también están limitadas a un rango determinado. Con todo lo anterior, se plantea el siguiente problema de optimización:

$$\max \quad Z = B \quad (2.6)$$

$$\text{s.a.} \quad -(G^\circ + R \cdot T \cdot S^T)x \geq B \quad (2.7)$$

$$\ln(C_{min}) \leq x \leq \ln(C_{max}) \quad (2.8)$$

Donde $\ln(C_{min})$ y $\ln(C_{max})$ son los logaritmos naturales las concentraciones mínima y máxima que puede tomar x , respectivamente. B representa el límite inferior de las fuerzas impulsoras de las reacciones involucradas. Entonces, al maximizar B todas las reacciones están lo más lejos posible del equilibrio, a este valor se le nombra *Max-min Driving Force (MDF)*. Con esto se obtienen cuáles son los cuellos de botella en todas las vías a analizar, que corresponden a las reacciones menos favorables y que, por ende, requerirán una mayor concentración de enzima para realizarse.

El método *MEMO (MEtabolic MOdules)*, o módulos metabólicos) se basa en estos límites termodinámicos utilizando *MDF* para diseñar y predecir rutas metabólicas [7]. Un supuesto fuerte que tiene este método es priorizar la ruta más corta en base a restricciones termodinámicas sin tomar en cuenta la cinética de las reacciones o la eficiencia de las enzimas involucradas. Comienza con un conjunto de reacciones disponibles a utilizar y un planteamiento similar al de un *FBA* para los balances de masa y el estado pseudo-estacionario, considerando todo el universo de reacciones como irreversibles. Para lograr esto último las reacciones reversibles se separan como 2 reacciones distintas y cada una con su respectivo conjunto factible. Luego, se establece para cada flujo la siguiente restricción:

$$LB_j \cdot z_j \leq v_j \leq UB_j \cdot z_j$$

Donde LB_j y UB_j son los límites inferiores y superiores, ambos positivos, y z_j es una variable binaria que toma valor 1 cuando por la reacción j hay un flujo y 0 en caso contrario. Esta variable también se encuentra presente en otras restricciones de la metodología *MDF*, al complementar las restricciones de ambos métodos se obtienen flujos mayores en las reacciones que sean termodinámicamente más favorables. Finalmente, escogiendo un sustrato inicial y un producto celular de interés, se minimiza la cantidad de reacciones activas necesarias para producir dicho compuesto.

Otro método de diseño reciente que incorpora la estequiometría de las reacciones como factor de decisión en el diseño de rutas metabólicas es *optStoic* [26]. En la Figura 2.6 se aprecia el *workflow* del algoritmo a utilizar, la cual se divide en dos pasos principales:

- **Paso 1:** Se busca encontrar la estequiometría óptima a partir de uno o más sustratos iniciales, uno o más productos finales, y restricciones para balances de masa, balances de carga y factibilidad termodinámica. Además, se determina la estequiometría de posibles co-reactantes y co-productos que no se hayan especificado en la entrada (C y D en la Figura 2.6). También pueden incluirse los costos del sustrato inicial y las potenciales ganancias del producto final si se consideran restricciones adicionales. Cabe destacar que esta corresponde a una estequiometría general.
- **Paso 2:** Se determina la combinación de reacciones óptima para lograr la estequiometría general obtenida en el paso 1. En este paso se pueden utilizar dos criterios; minimizar el número de reacciones (*minRxn*) o minimizar la suma total de flujos a través de las reacciones (*minFlux*).

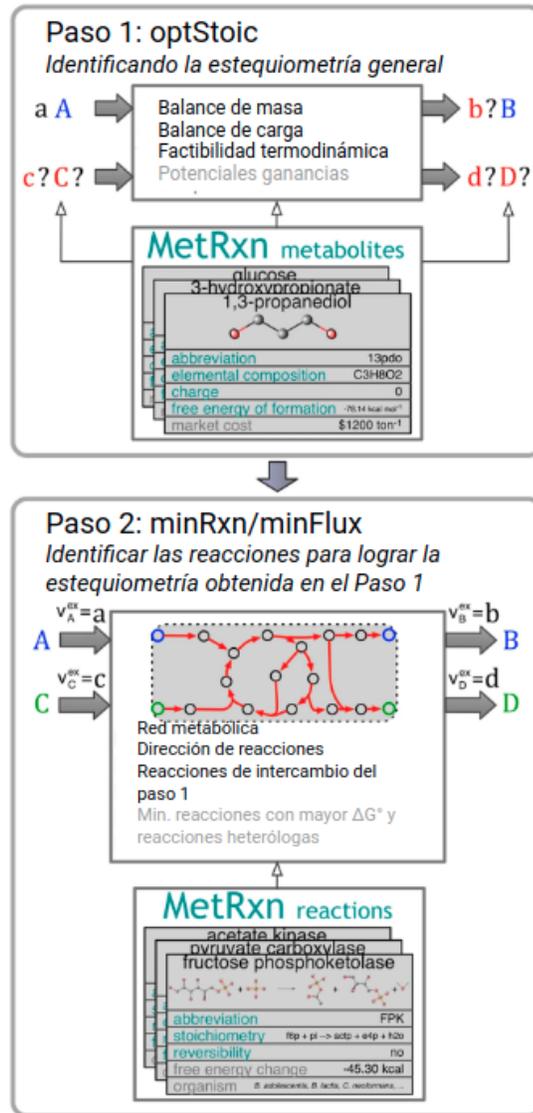


Figura 2.6: Representación gráfica del método *optStoic*, adaptado de Chowdhury et al. (2015)

El método *Enzyme Cost Minimization (ECM)* se desarrolló para estudiar y analizar el costo enzimático que tiene una determinada ruta considerando la cinética de las reacciones involucradas [24], entendiéndose como costo enzimático la cantidad de enzima necesaria para lograr un cierto flujo. Cabe destacar que este método se puede utilizar sólo para analizar una ruta definida, no para el diseño de la misma. Por otro lado, incorpora parte del método *MDF* en un factor de reversibilidad pero también utiliza distintas simplificaciones para todas las reacciones como una cinética reversible de Michaelis-Menten y los factores utilizados para evaluar la saturación y la regulación corresponden a las expresiones de una inhibición alostérica no competitiva, lo cual puede no ser cierto para distintas reacciones, sobretodo las que involucran producción o consumo energético.

Este método consiste principalmente en el planteamiento de dos variables, una describe

la velocidad de reacción de la enzima v y otra se define como el costo enzimático q , ambas variables se pueden apreciar en la Figura 2.7. La primera ecuación describe la velocidad de reacción de la enzima en función de los niveles de enzima (E), su constante catalítica en el sentido directo de la reacción (k_{cat}^+), un término asociado a la reversibilidad (η^{rev}) y otro factor que engloba la saturación y la regulación de la misma enzima ($\eta^{kin} = \eta^{sat} \cdot \eta^{reg}$). A partir de esta ecuación y los parámetros mencionados se calcula E . La carga de la enzima, presente en la segunda ecuación, es una variable que representa todos los recursos y procesos que requiere una enzima para catalizar una reacción, incluyendo sus modificaciones postraduccionales, mantención de la enzima en sí, los efectos de un mal plegamiento y catálisis no específica. Este término se estima según el tamaño de la enzima, composición de aminoácidos, tiempo de vida medio, etc. Finalmente, con el nivel y la carga de la enzima se obtiene el costo enzimático.

$$\begin{array}{c}
 \begin{array}{cccccc}
 \text{velocidad de} & \text{nivel de} & \text{constante} & \text{factor de} & \text{factor de} & \text{factor de} \\
 \text{reacción} & \text{enzima} & \text{catalítica} & \text{reversibilidad} & \text{saturación} & \text{regulación} \\
 & & \text{directa} & & & \\
 v & = & E & \cdot & k_{cat}^+ & \cdot & \underbrace{[1 - e^{-\theta}]}_{\eta^{rev}} & \cdot & \underbrace{\frac{s/K_S}{1 + s/K_S + p/K_P} \cdot \frac{1}{1 + x/K_I}}_{\eta^{kin}}
 \end{array} \\
 \\
 \begin{array}{c}
 \text{costo} \\
 \text{enzimático} \\
 \swarrow \\
 q = h_E \cdot E = h_E \cdot v \cdot \frac{1}{k_{cat}^+} \cdot \underbrace{\frac{1}{[1 - e^{-\theta}]}}_{1/\eta^{rev}} \cdot \underbrace{\frac{1 + s/K_S + p/K_P}{s/K_S} \cdot [1 + x/K_I]}_{1/\eta^{kin}}
 \end{array}
 \end{array}$$

Figura 2.7: Ecuaciones principales del método *ECM*, adaptado de Noor et al. (2016)

Para combinar los dos últimos métodos nombrados, los autores de *optStoic* publicaron su segunda versión donde buscaban predecir rutas alternativas a la glicólisis y posteriormente evaluar su respectivo costo enzimático [27]. Como el objetivo de esta segunda versión de *optStoic* era buscar alternativas a una ruta conservada, el paso 1 de la Figura 2.6 fue reemplazado por una imposición de los límites para los flujos de entrada y salida del sustrato inicial, el producto final y los posibles co-reactantes o co-productos. Las predicciones en sí no influyen en el método *ECM*, sino que este es usado luego de obtener las rutas alternativas para evaluar el costo enzimático.

Capítulo 3

Objetivos

El objetivo principal de este trabajo es diseñar e implementar un algoritmo que prediga vías metabólicas con carga enzimática mínima y que sea aplicable a cualquier caso de interés, independiente del microorganismo hospedero a utilizar, el sustrato inicial o el producto deseado.

Como objetivos específicos se tienen:

- Obtener las constantes catalíticas y actividades específicas disponibles de reacciones biológicas en **BRENDA**.
- Modelar la carga enzimática a través de un problema de optimización utilizando los parámetros cinéticos obtenidos y un modelo metabólico ya existente.
- Validar el modelo con datos experimentales obtenidos de bibliografía.
- Plantear o buscar en bibliografía un problema de optimización que diseñe una ruta de producción.
- Incorporar la minimización de la carga enzimática al problema anterior a través de las restricciones del problema.
- Validar el desempeño del algoritmo en tres casos de estudio: una red metabólica con reacciones y parámetros ficticios, una red que represente el metabolismo central donde se estudien rutas conocidas con parámetros reales ya extraídos, y una red metabólica universal en la cual se busque diseñar vías alternativas para la producción de compuestos isoprenoides, utilizando también parámetros reales.

Capítulo 4

Metodología

4.1. Obtención de parámetros cinéticos

Previo al planteamiento del algoritmo, se realizó la extracción de parámetros cinéticos necesarios para la aplicación del método. Estos se encuentran disponibles en la base de datos BRENDA [34] y se clasifican según el *EC number*, un esquema de clasificación numérica para las enzimas. Este esquema consta de cuatro números donde cada uno especifica cierta funcionalidad de la enzima. El primer número corresponde a una caracterización general y cada número a su derecha aporta información cada vez más específica.

Se muestra como ejemplo la enzima alcohol deshidrogenasa:

- 1: oxido reductasa
- 1.1: actúa en los grupos donantes $CH - OH$
- 1.1.1: con NAD^+ o $NADP^+$ como aceptor
- 1.1.1.1: alcohol deshidrogenasa que utiliza NAD^+
- 1.1.1.2: alcohol deshidrogenasa que utiliza $NADP^+$

La extracción se realizó utilizando expresiones regulares en el lenguaje de programación `python` con la librería `re`. Se obtuvieron la constante catalítica y la actividad específica de las enzimas disponibles. Además, se obtuvieron para su posterior clasificación las condiciones óptimas y rangos de pH y temperatura en las que funciona cada enzima reportada.

Del universo de enzimas presente en BRENDA, se consideraron aquellas presentes de todos los organismos disponibles que correspondan a cepas no mutantes, que la temperatura de funcionamiento sea menor a los 40°C y el pH de funcionamiento esté entre 4.0 y 8.0. Este filtro fue aplicado para no considerar condiciones extremas de temperatura o pH ni enzimas modificadas, así se evita utilizar información de enzimas puntuales que no están presentes en las condiciones de operación típicas para la producción recombinante a gran escala.

Para las enzimas que según su *EC number* no cuenten con la información requerida, se utilizará como valor representativo la media de los logaritmos de la actividad específica de

todas las enzimas que comparten los primeros tres números del *EC number*. Como comparten tres de las cuatro clasificaciones, se tomó el supuesto de que sus parámetros deben ser similares debido a que su función es similar. Para las enzimas en que se tenía más de un dato de constante catalítica o actividad específica se utilizó también el promedio de los logaritmos.

4.2. Modelamiento de la carga enzimática

Como primer acercamiento, se buscó evaluar la carga enzimática de una red completa sin buscar diseñar una nueva ruta. Para ello se plantea el siguiente problema de optimización:

$$\min \quad z = \sum_{j=1}^n E_j \quad (4.1)$$

$$\text{s.a.} \quad E_j \in \mathbb{R}^+ \quad , \forall j = 1, \dots, n \quad (4.2)$$

$$S \cdot v = 0 \quad (4.3)$$

$$\mu = \mu_{opt} \quad (4.4)$$

$$E_j \cdot K_j^- \leq v_j \leq E_j \cdot K_j^+ \quad , \forall j = 1, \dots, n \quad (4.5)$$

Donde:

- E_j corresponden a las cargas enzimáticas para cada reacción.
- μ_{opt} es el flujo óptimo para la biomasa, el cual se obtiene resolviendo el *FBA* respectivo.
- K_j^- y K_j^+ corresponden a las actividades específicas de las enzimas asociadas a cada reacción. En este caso, el parámetro obtenido de **BRENDA** corresponde a K_j^+ , mientras que K_j^- se obtiene a partir de la ecuación 4.6, que relaciona la constante de equilibrio de una reacción con la energía libre de Gibbs [35]. La constante de equilibrio se define como el cociente entre la constante cinética en el sentido directo de la reacción K^+ y la constante cinética en el sentido inverso K^- .

$$\Delta G_{rxn} = -R \cdot T \ln(K_{eq}), \quad K_{eq} = \frac{K^+}{K^-} \quad (4.6)$$

$$K^- = \frac{K^+}{\exp^{-\Delta G/R \cdot T}} \quad (4.7)$$

Este problema de optimización fue implementado en **python** utilizando la librería **cobrapy** [19] y se evaluó la carga enzimática de dos *GSM* de *Escherichia coli* disponibles en **BiGG Models**. Uno de ellos es **e coli core**, que involucra solo las rutas pertenecientes al metabolismo central. También se utilizó el modelo **iJO1366**, que considera todo el metabolismo de *E. coli* [9]. Posteriormente, se validó el modelo comparando con datos experimentales obtenidos de bibliografía. En el estudio utilizado se cuantificaron los flujos intracelulares para distintos cultivos de la misma cepa y en iguales condiciones pero a cada cultivo se le realizó un *knock out* distinto [32].

4.3. Elección del método base para el desarrollo del algoritmo e incorporación de la carga enzimática

De los métodos comentados en la sección 2.3, se elige la segunda versión de *optStoic* como el método base para este trabajo [27]. Este método ya existente cuenta con distintas ventajas por las que es escogido como punto de partida:

- En sus inicios fue pensado para diseñar rutas *de novo*.
- Su implementación es de libre uso, distribución y modificación.
- Se encuentra implementado en **python**, por lo que es más fácil de acoplar con el trabajo realizado anteriormente al incorporar nuevas restricciones.
- Es capaz de obtener más de una ruta resultante.
- Es un método relativamente actual ya que fue publicado hace menos de dos años.
- Cuenta con un universo de reacciones depurado por los mismos autores que contempla su clasificación según reversibilidad, se especifica si son reacciones de transporte o no y se identificaron todos los posibles *loops* que podrían formarse en una solución para la generación o consumo de *ATP* y transportadores de electrones. Para evitar la formación de *loops* en las soluciones se consideran restricciones adicionales.

Se utilizó este universo de reacciones para el caso de estudio de los compuestos isoprenoides combinado a los parámetros obtenidos de **BRENDA**.

Como se vio en la figura 2.6, este método fue pensado inicialmente como dos problemas de optimización. El primero identifica la estequiometría general óptima a partir de un sustrato, un producto deseado, balances de masa, balances de carga, factibilidad termodinámica y posibles ganancias según la producción. El segundo encuentra la ruta óptima para lograr dicha estequiometría general. Considerando el mismo ejemplo que en la metodología de esta primera implementación ($aA + cC \rightarrow bB + dD$), el planteamiento de ambos problemas de optimización es el siguiente [26]:

$$\max \quad z = f(a, b, c, d) \quad (4.8)$$

$$\text{s.a.} \quad n_{Bq} \cdot b + n_{Dq} \cdot d - n_{Aq} \cdot a - n_{Cq} \cdot c = 0 \quad , \forall q \in Q \quad (4.9)$$

$$e_B \cdot b + e_D \cdot d - e_A \cdot a - e_C \cdot c = 0 \quad (4.10)$$

$$\Delta G_B^f \cdot b + \Delta G_D^f \cdot d - \Delta G_A^f \cdot a - \Delta G_C^f \cdot c \leq -\Delta G^{min} \quad (4.11)$$

$$pr_B \cdot b + pr_D \cdot d - pr_A \cdot a - pr_C \cdot c \geq profit^{min} \quad (4.12)$$

$$h(a, b, c, d) = 1 \quad (4.13)$$

$$a, b, c, d \in \mathbb{R} \quad (4.14)$$

Donde:

- a, b, c, d son los coeficientes estequiométricos de los reactivos y productos a considerar, cabe destacar que los co-reactantes (b) y co-productos (d) pueden no estar definidos.
- f es la función que define el uso del reactivo para la conversión a producto, es elegida por el usuario y depende del caso de estudio. Esta conversión puede ser completa, co-utilización de un reactante A para generar dos productos B y D, entre otras opciones.
- n_{iq} es el número de átomos del elemento q en el compuesto i .
- e_i : carga del compuesto i .
- ΔG_i^f : es la energía libre de Gibbs a $25^\circ C$ y fuerza iónica 0.1 M.
- ΔG^{min} es la cantidad de energía libre extra que debe estar disponible para lograr la estequiometría deseada.
- pr_i es el valor monetario del compuesto i , para los reactantes corresponde al costo mientras que para los productos corresponde a la ganancia.
- $profit^{min}$ es la utilidad mínima que se desea obtener.
- h : es una función de escala en caso de requerir normalizar los coeficientes estequiométricos.

$$\min \quad z = \sum_{j \in J/J_{exchange}} v_{f,j} + v_{b,j} \quad (4.15)$$

$$\text{s.a.} \quad v_j = v_{f,j} - v_{b,j} \quad (4.16)$$

$$\sum_{j \in J} S_{i,j} \cdot v_j = 0, \quad \forall i \in I \quad (4.17)$$

$$v_i^{EX} = q_i, \quad \forall i \in I_{stoich} \quad (4.18)$$

$$v_{f,j} \geq y_{f,j} \cdot 0.5, \quad \forall j \in J \quad (4.19)$$

$$v_{f,j} \leq y_{f,j} \cdot M, \quad \forall j \in J \quad (4.20)$$

$$v_{b,j} \geq y_{b,j} \cdot 0.5, \quad \forall j \in J \quad (4.21)$$

$$v_{b,j} \leq y_{b,j} \cdot M, \quad \forall j \in J \quad (4.22)$$

$$v_j \in \mathbb{Z} \vee \mathbb{R}, \quad v_i^{EX} \in \mathbb{Z} \vee \mathbb{R} \quad (4.23)$$

$$\sum_j N_{red,jl}^T G_j = 0, \quad \forall l \in L \quad (4.24)$$

$$G_j \geq -M \cdot a_j + (1 - a_j), \quad \forall j \in J \quad (4.25)$$

$$G_j \geq -a_j + M \cdot (1 - a_j), \quad \forall j \in J \quad (4.26)$$

$$v_j \geq -M \cdot (1 - a_j), \quad \forall j \in J \quad (4.27)$$

$$v_j \leq M \cdot a_j, \quad \forall j \in J \quad (4.28)$$

$$G_j \in \mathbb{R}, \quad a_j \in \{0, 1\} \quad (4.29)$$

Donde:

- $v_{f,j}$ es el flujo que pasa por la reacción j en el sentido directo ($A \rightarrow B$).
- $v_{b,j}$ es el flujo que pasa por la reacción j en el sentido inverso ($A \leftarrow B$).
- v_j es el flujo neto que pasa por la reacción j , se calcula como la resta entre los flujos de la reacción directa e inversa.
- $S_{i,j}$ es la matriz estequiométrica que representa el universo de reacciones a considerar, donde i corresponde a los metabolitos y j a las reacciones.
- v_i^{EX} es el flujo del compuesto i que corresponde a las reacciones de intercambio, tomando el valor q_i , según la estequiometría impuesta por el usuario.
- $y_{f,j}$ e $y_{b,j}$ son parámetros de encendido y apagado para la reacción j en el sentido directo e inverso, respectivamente.
- $N_{red,jl}^T$ es la matriz que contiene todos los *loops* de generación y consumo de cofactores en las reacciones internas de la red metabólica (sin entradas ni salidas de la estequiometría general impuesta) para la reacción j y el *loop* l .
- G_j es una variable de pseudo energía libre, esta variable no es la fuerza impulsora de la reacción j ($\Delta_r G'_j$) pero busca representarla sin que sea necesario conocer el parámetro real.
- M es un valor positivo grande (1000).
- a_j son variables binarias que toman el valor 1 cuando hay flujo por la reacción j y 0 en caso contrario. Estas variables ayudan a relacionar las restricciones termodinámicas con la minimización de flujos.
- I es el conjunto de todos los compuestos de la red metabólica.
- I_{stoich} es el conjunto de compuestos de entrada y salida de la red que es definida por la estequiometría general impuesta.
- J es el conjunto de todas las reacciones de la red metabólica.
- L es el conjunto de todos los *loops* posibles en la red metabólica.

En el planteamiento de la segunda versión de *optStoic* no se considera el primer paso, sino que solo se busca la ruta metabólica para una estequiometría definida por el usuario. Esto se debe a que para esta segunda implementación se buscaron alternativas a rutas conservadas del metabolismo central en lugar de hacer un diseño *de novo* [27]. Para este trabajo también se entregará una estequiometría particular para la minimización de la carga enzimática, dejando como trabajo futuro la implementación del primer problema de optimización que obtiene la estequiometría general óptima.

Finalmente, la nueva función objetivo queda expresada en la ecuación 4.30 donde se busca minimizar la carga enzimática. Se agregaron a las restricciones que limitan el flujo de las reacciones en sentido directo e inverso (Ecuaciones 4.31 y 4.32, respectivamente) y las restricciones de encendido y apagado de las reacciones (Ecuaciones 4.33 y 4.34, respectivamente)

según la reversibilidad de las reacciones que está dada por los parámetros binarios $y_{f,j}$ y $y_{b,j}$. Estos parámetros no son variables del problema de optimización, sino que forman parte del universo de reacciones de *optStoic* y representan si hay flujo por la reacción j . Los factores numéricos de las Ecuaciones 4.31 y 4.32 corresponden a conversiones de unidades de minutos a horas.

$$\min \sum_{j \in J} E_{f,j} + E_{b,j} \quad (4.30)$$

$$v_{f,j} \leq E_{f,j} \cdot K_j^+ \cdot 60 \quad (4.31)$$

$$v_{b,j} \leq E_{b,j} \cdot K_j^- \cdot 60 \quad (4.32)$$

$$E_{f,j} \leq y_{f,j} \cdot M \quad (4.33)$$

$$E_{b,j} \leq y_{b,j} \cdot M \quad (4.34)$$

Es importante notar que al tener información cinética para una reacción en ambos sentidos se consideran distintas cargas enzimáticas para cada sentido, dándole un sentido biológico al minimizar la suma de las cargas en el sentido directo e inverso en lugar de considerar solo la carga enzimática neta $E_j = E_{f,j} - E_{b,j}$. Si se considerara la resta de ambas cargas enzimáticas como objetivo a minimizar, el optimizador podría darle valores muy grandes a ambas cargas y anular su resta, llegando a un óptimo que no se podría interpretar correctamente con lo que ocurre dentro de la célula.

Una vez implementado el algoritmo, se pondrá a prueba en 3 escenarios distintos. Primero, una red metabólica ficticia que permita depurar la implementación. Luego, una red metabólica que contemple el metabolismo central donde se estudien rutas conocidas con los parámetros previamente extraídos. Finalmente, el universo de reacciones que venía incorporado en *optStoic* combinado con los parámetros de **BRENDA** para estudiar la producción de *IPP*, precursor principal de los compuestos isoprenoides.

Capítulo 5

Resultados

5.1. Evaluación de la carga enzimática en un *GSM*

El código fuente desarrollado para la extracción de parámetros de BRENDA y el posterior filtro de dichos parámetros se puede apreciar en el Anexo 8.1. Los valores obtenidos para las constantes catalíticas y actividades específicas filtradas según pH, temperatura y procedencia de organismos silvestres se puede apreciar en la Tabla 5.1.

Tabla 5.1: Resumen de los parámetros cinéticos filtrados

Parámetro estadístico	Constante catalítica $\left[\frac{\text{mol Enzima}}{\text{mol sustrato} \cdot \text{seg}} \right]$	Actividad específica $\left[\frac{\text{mmol Sustrato}}{\text{gr Enzima} \cdot \text{min}} \right]$
Datos totales	7294	7294
Promedio	2.12	25.45
Desviación estándar	12.42	324.34
Mínimo	$7 \cdot 10^{-5}$	$1 \cdot 10^{-6}$
25 %	0.44	0.28
50 %	0.73	1.16
75 %	0.90	3.69
Máximo	642.9	18300

Cada parámetro muestra una gran dispersión, la desviación estándar es 6 veces mayor que el promedio en el caso de la constante catalítica y 13 veces mayor en el caso de la actividad específica. Sin embargo, esto es de esperarse, pues las enzimas existentes que han sido reportadas varían considerablemente tanto en sus funciones como en su eficiencia. También se aprecia que entre el primer y el tercer cuartil los valores de ambos parámetros mantienen sus órdenes de magnitud. Es bajo el primer cuartil y sobre el tercer cuartil donde hay valores de órdenes de magnitud dispares. Estos valores son los responsables de la alta desviación estándar obtenida en cada conjunto de datos.

Se decidió continuar con el uso de las actividades específicas por sobre las constantes catalíticas para evitar obtener moles de enzima en las unidades de los flujos al aplicar el algoritmo, prefiriendo trabajar con gramos de enzima directamente y no necesitar los pesos

moleculares de cada enzima para realizar esta conversión.

Luego, se calculó la constante respectiva a cada reacción en el sentido inverso aplicando la ecuación 4.7 con las actividades específicas y los ΔG_{rxn} , que fueron obtenidos previamente a este trabajo utilizando el método de contribución de grupos y el método de contribución de componentes [36]. En la Tabla 5.2 aparece la caracterización estadística de ambos parámetros.

Tabla 5.2: Resumen de los ΔG_{rxn} y actividades específicas inversas. (***):
por la diferencia de órdenes de magnitud de los datos no se logra calcular
la desviación estándar de este parámetro.

Parámetro estadístico	ΔG_{rxn} $\left[\frac{kJ}{mol}\right]$	Act. específica inversa $\left[\frac{mmol\ Sustrato}{gr\ Enzima \cdot min}\right]$
Datos totales	5757	5751
Promedio	-43.02	$2.56 \cdot 10^{277}$
Desviación estándar	144.30	***
Mínimo	-1332.0	0
25 %	-24.18	$6.67 \cdot 10^{-10}$
50 %	-0.80	0.06
75 %	0.0	4.32
Máximo	4568.30	$14.75 \cdot 10^{277}$

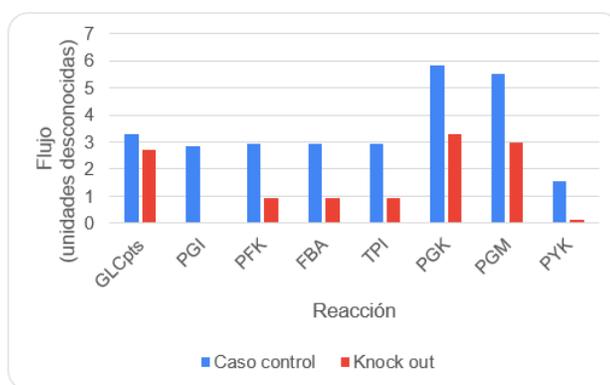
Los valores de las constantes inversas a la actividad específica resultaron ser demasiado dispares, esto se debe principalmente a la gran dispersión de los ΔG_{rxn} . Si bien este resultado es de esperarse al igual que con las constantes extraídas de BRENDA, esta gran dispersión sumada a la presencia de una exponencial en la ecuación 4.7 genera la divergencia presentada en la Tabla 5.2 para las actividades específicas inversas. Esto va a representar un desafío a la hora de plantear la carga enzimática, por lo que en el futuro se propone filtrar nuevamente por un rango aceptable de valores para que el optimizador no colapse por la diferencia de órdenes de magnitud.

Luego de la extracción de parámetros y plantear la carga enzimática como el problema de optimización descrito en la sección 4.2 (implementación en Anexo 8.2), se buscó validar los resultados obtenidos de estos modelos. Para ello se compararon flujos obtenidos experimentalmente con *C-based metabolic flux analysis*, un enfoque para cuantificar la respuesta de redes metabólicas en microorganismos. Esto se hace complementando un análisis teórico como los vistos en la sección 2.3 con distintas técnicas de laboratorio que cuantifican el *ARNm* de las proteínas involucradas y los metabolitos involucrados en la ruta, tales como *qRT – PCR* y electroforesis capilar acoplada a espectrometría de masas [31].

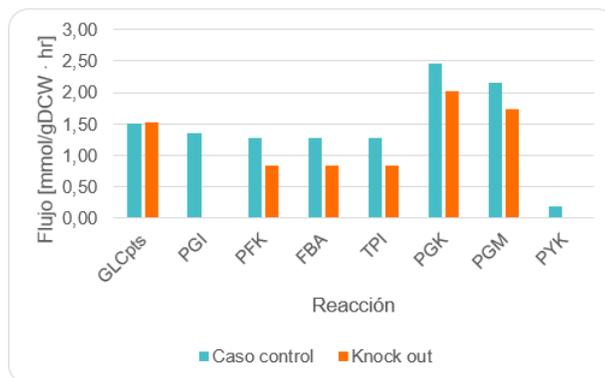
En dicha publicación se trabaja con *Escherichia coli* cultivada en un quimiostato a una tasa de dilución $D = 0,2[hr^{-1}]$ y se realiza la cuantificación de los flujos intracelulares a distintas muestras. Se cuenta con una cepa control sin modificaciones en su genoma y otras cepas a las cuales se les ha silenciado un gen en particular asociado a la glicólisis o a la ruta de las pentosas fosfato [32]. Por la complejidad de esta evaluación se consideró sólo el modelo *E. coli core*, pero se realizaron los *knock-out* respectivos al *GSM* para cada caso estudiado.

Se realizaron cinco simulaciones con sus respectivos genes silenciados. Se compararon cualitativamente los resultados del modelo y los datos de la publicación usada como referencia. Por la cantidad de flujos medidos, ensayos experimentales realizados y las limitaciones de comparación con los datos experimentales, se muestran solo los flujos de las reacciones que tienen asociadas alguno de los genes deletados, los cuales participan en la ruta de la glicólisis. Cabe destacar que los resultados del modelo corresponden a flujos calculados en $\left[\frac{\text{mmol}X}{\text{gDCW}\cdot\text{hr}^{-1}}\right]$, acorde a las unidades de flujo que se utilizan en los modelos metabólicos de BiGG, mientras que los resultados de la publicación fueron normalizados por la tasa de absorción de glucosa de cada cultivo. Como esta tasa no se encuentra disponible en el material suplementario de la publicación ni se especifican las unidades finales de la cuantificación, solo se compara el aumento o disminución de los flujos estudiados al no poder igualar las unidades.

En las Figuras 5.1 y 5.2 puede observarse la comparación de dos *knock out* estudiados. Por la similitud de los demás gráficos obtenidos, las otras deletaciones estudiadas se encuentran en el Anexo 8.3.

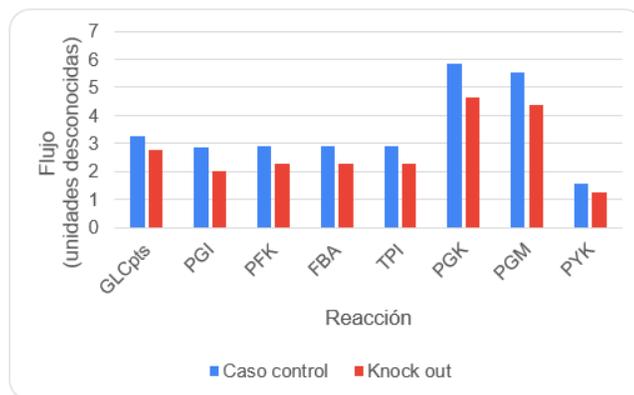


(a) Datos experimentales

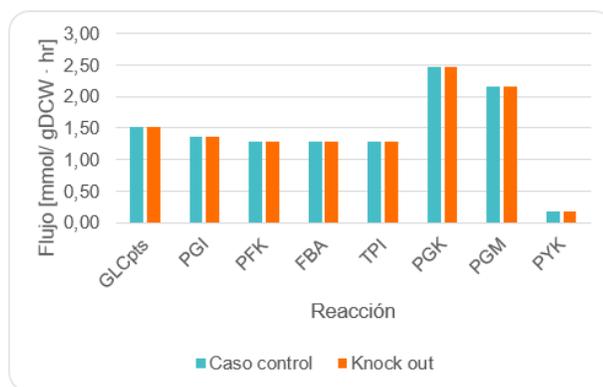


(b) Simulación

Figura 5.1: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b4025 (proteína PGI)



(a) Datos experimentales



(b) Simulación

Figura 5.2: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b2097 (proteína fbaB)

En la Figura 5.1 se puede observar que el flujo asociado al gen deletado es nulo en ambos casos y que, excepto por la reacción GLCpts, todos los flujos disminuyen frente al *knock out* especificado con respecto al caso control, tanto en la predicción del modelo como en los datos experimentales. De esto se puede inferir que se llega de todas formas a una distribución de flujos similar a la obtenida de forma experimental frente a la misma deleción aún con las modificaciones hechas al *FBA* para incorporar la carga enzimática, las nuevas variables agregadas al sistema y las restricciones más acotadas. La disminución de todas las reacciones afectadas es alrededor de un 50%, esto se debe a que el *knock out* realizado corresponde a la segunda reacción de la glicólisis, cortando el flujo para todas las reacciones siguientes que son de la misma ruta.

En el caso de la Figura 5.2 y las demás deleciones estudiadas (ver Anexo 8.3) no se aprecian cambios en la distribución de flujos obtenida por el modelo, pero en los datos experimentales se observa una variación, aunque esta es menor que la obtenida en la Figura 5.1. Esto se debe a que dicha reacción tiene más de un gen codificante para una enzima asociada a esta reacción. Al producir el *knock out*, la reacción puede seguir siendo catalizada, solo que a una tasa más baja y no se anula dicho flujo como en el primer caso mencionado. Este comportamiento se aprecia en los datos experimentales, pero no en la predicción del modelo, principalmente

porque en la resolución el modelo asume que al tener más de un posible camino para llegar al óptimo deseado, no importa si un gen fue deletado mientras otro esté disponible y no requiera de la acción conjunta de ambos genes, es posible catalizar la reacción a la misma tasa. Por ende, el caso óptimo es en el que este gen no deletado codifica para la enzima en la cantidad necesaria para lograr el mismo flujo y no variar la distribución general.

Es importante notar que aunque se utiliza un *GSM* para estas simulaciones, el modelamiento de la carga enzimática no contiene en su planteamiento regulaciones a nivel del genoma, sino que se basa en las reacciones presentes en el modelo metabólico e impone límites para los flujos de estas reacciones. En este caso se hace uso de la asociación entre genes y proteínas que ya está incorporada en el *GSM* para evaluar cómo afectan estos nuevos límites de los flujos a la distribución final obtenida en distintos escenarios.

Con el trabajo realizado hasta este punto se buscó modelar la carga enzimática utilizando un problema de optimización similar al *FBA* que incorporara los parámetros extraídos en las restricciones de flujo. En términos generales, se logró obtener una distribución de flujos y ésta es similar a la esperada según los datos de bibliografía. En la próxima sección se mostrará cómo se incorporó este planteamiento a *optStoic*, un método ya existente que fue pensado para el diseño de rutas metabólicas.

5.2. Desempeño del algoritmo

El desarrollo del algoritmo fue realizada a partir del código fuente de *optStoic*, la implementación en detalle se encuentra en el Anexo 8.4. Una vez incorporada la opción de minimizar la carga enzimática a *optStoic*, para evaluar el desempeño del algoritmo se consideraron 3 conjuntos de reacciones:

- Una red metabólica de 7 reacciones ficticias, tanto sus compuestos como sus actividades específicas.
- Una red metabólica que contempla el metabolismo central con actividades específicas reales.
- El universo de reacciones incluido en *optStoic*, también con valores reales para las actividades específicas.

El universo de reacciones que viene incorporado en *optStoic* consta de 6359 reacciones, de las cuales se obtuvo la actividad específica para 3826 de ellas a partir de los datos extraídos y filtrados de **BRENDA**. Sin embargo, para evitar problemas de convergencia se consideraron solo actividades específicas que fluctúen entre $1 \cdot 10^{-9}$ y $1 \cdot 10^9$, dejando un total de 2805 reacciones. Para las reacciones que no se tenga información, se mantienen las cotas predefinidas para minimizar flujos y no se considerarán variables de carga enzimática asociadas a ellas.

5.2.1. Prueba de concepto

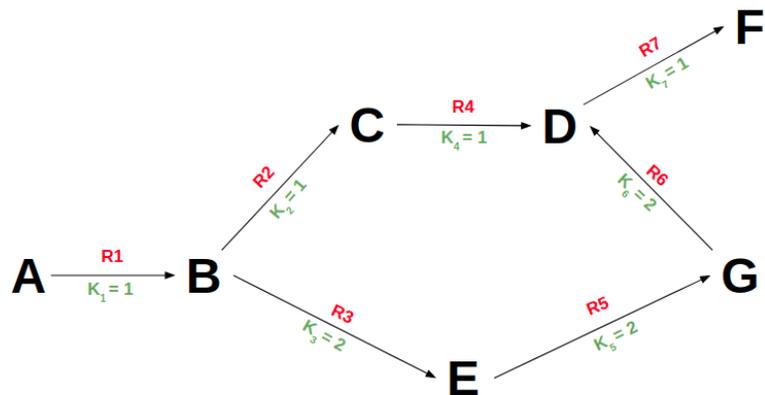
En la Figura 5.3(a) se aprecia la red metabólica que se utilizó para comprobar el correcto funcionamiento del algoritmo en desarrollo. En esta prueba de concepto existen dos rutas para llegar desde el sustrato inicial A al producto final F, la ruta A-B-C-D-F corresponde a la ruta con menor flujo, mientras que la ruta A-B-E-G-D-F posee menor carga enzimática. En la Tabla 5.3 aparecen los límites impuestos al algoritmo para lograr obtener las rutas mencionadas, donde A al ser el sustrato inicial aparece con signo negativo para forzar su consumo y F con signo positivo para forzar su producción.

Tabla 5.3: Rangos de tasas de producción o consumo para los compuestos de entrada y salida, prueba de concepto

Compuesto	$LB(\frac{mmol}{gr S_0 \cdot hr})$	$UB(\frac{mmol}{gr S_0 \cdot hr})$
A	-1	-1
F	1	1

El método desarrollado permite obtener distintas soluciones a los casos presentados. Dichas soluciones se entregan de forma ordenada de acuerdo a la función objetivo, siendo la primera la solución óptima y la o las siguientes alternativas subóptimas. Al minimizar la carga enzimática se obtienen dos soluciones, primero la ruta de la Figura 5.3(b) y luego la de la Figura 5.3(c), mientras que al minimizar flujos se obtienen las mismas rutas en el orden contrario. Este resultado es satisfactorio, pues se cumple lo esperado tanto al minimizar flujo como al minimizar carga enzimática. Se aprecia en la Tabla 5.4 que la carga enzimática disminuye a la mitad en las reacciones que tienen como actividad específica $k_j = 2$ en relación a las reacciones con $k_j = 1$.

Es importante notar que esta prueba de concepto fue diseñada como un caso simple para verificar el correcto funcionamiento del algoritmo, por esto es que no considera casos más complejos como una tercera vía para producir el compuesto deseado, no incluye *loops*, formación o consumo de ATP, uso de transportadores de electrones o reacciones reversibles.

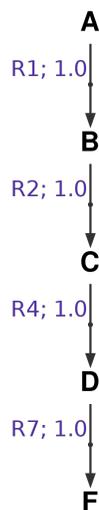


(a) Red metabólica de prueba, elaboración propia: en negro se representan los metabolitos de la red, en rojo el ID de la reacción y en verde la actividad específica correspondiente.

toy_model_pathway_1



toy_model_pathway_2



(b) Ruta obtenida minimizando carga enzimática (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).

(c) Ruta obtenida minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).

Figura 5.3: Resultados prueba de concepto: producción de F a partir de A

Tabla 5.4: Cargas enzimáticas para producir F a partir de A

Solución	ID reacción	Carga enzimática ($\cdot 10^3$)
Ruta 1	R1	16.6
	R3	8.33
	R5	8.33
	R6	8.33
	R7	16.6
	Carga total	58.1
Ruta 2	R1	16.6
	R2	16.6
	R4	16.6
	R7	16.6
	Carga total	66.4

5.2.2. Metabolismo central

Llevando el algoritmo a un escenario más complejo de forma progresiva, se elabora este conjunto de reacciones que involucran solo rutas conservadas: la glicólisis, el ciclo de Krebs y la ruta de las pentosas fosfato, para las cuales se dispone información de la actividad específica real. Con esta red se aumenta el grado de complejidad al que es sometido el algoritmo en cuanto a la cantidad de reacciones y el manejo de las actividades específicas con valores más diversos entre sí, pero manteniendo el conocimiento de las rutas posibles que el algoritmo puede tomar. En el Anexo 8.5 se especifica cómo fue elaborada esta red metabólica y qué reacciones incluye. El desempeño del algoritmo se evaluó en tres casos distintos para este conjunto de reacciones:

- producción de piruvato a partir de glucosa
- producción de gliceraldehído 3-fosfato a partir de glucosa
- producción de piruvato a partir de distintas fuentes de carbono (glucosa, malato y lactato)

El objetivo es obtener rutas que son conocidas del metabolismo central, pero donde puede variar qué ruta es elegida o qué sustrato inicial se utiliza para un determinado producto, además de evaluar cuál es más favorable según su flujo total y su carga enzimática. Para el primer caso de estudio se obtuvo como único resultado la glicólisis al minimizar flujos y carga enzimática, el detalle de este caso se encuentra en el Anexo 8.6.1.

5.2.2.1. Producción de gliceraldehído 3-fosfato a partir de glucosa

Este caso de estudio busca obtener dos rutas conocidas que comparten un punto de partida y de llegada, las cuales corresponden a las primera parte de la glicólisis y la ruta de las pentosas fosfato. Para ello se fijan los límites de entrada y salida adecuados para obtener ambas rutas mencionadas, dando más de libertad en los distintos cofactores. Los límites impuestos aparecen en la Tabla 5.5.

Tabla 5.5: Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de gliceraldehído 3-fosfato a partir de glucosa

Compuesto	$LB(\frac{mmol}{gr S_0 \cdot hr})$	$UB(\frac{mmol}{gr S_0 \cdot hr})$
Glucosa	-1	-1
Gliceraldehído 3-fosfato	1	2
Sedoheptulosa 7-fosfato	0	1
<i>ATP</i>	-4	4
<i>ADP</i>	-4	4
<i>NADP</i> ⁺	-4	4
<i>NAPDH</i>	-4	4
<i>H</i> ⁺	-7	7
<i>H</i> ₂ <i>O</i>	-2	2
<i>CO</i> ₂	-2	2

Los resultados obtenidos se pueden apreciar en la Figura 5.4 y sus cargas enzimáticas aparecen en la Tabla 5.6. Tanto al minimizar flujos como carga enzimática se obtienen las mismas dos rutas esperadas y en el mismo orden.

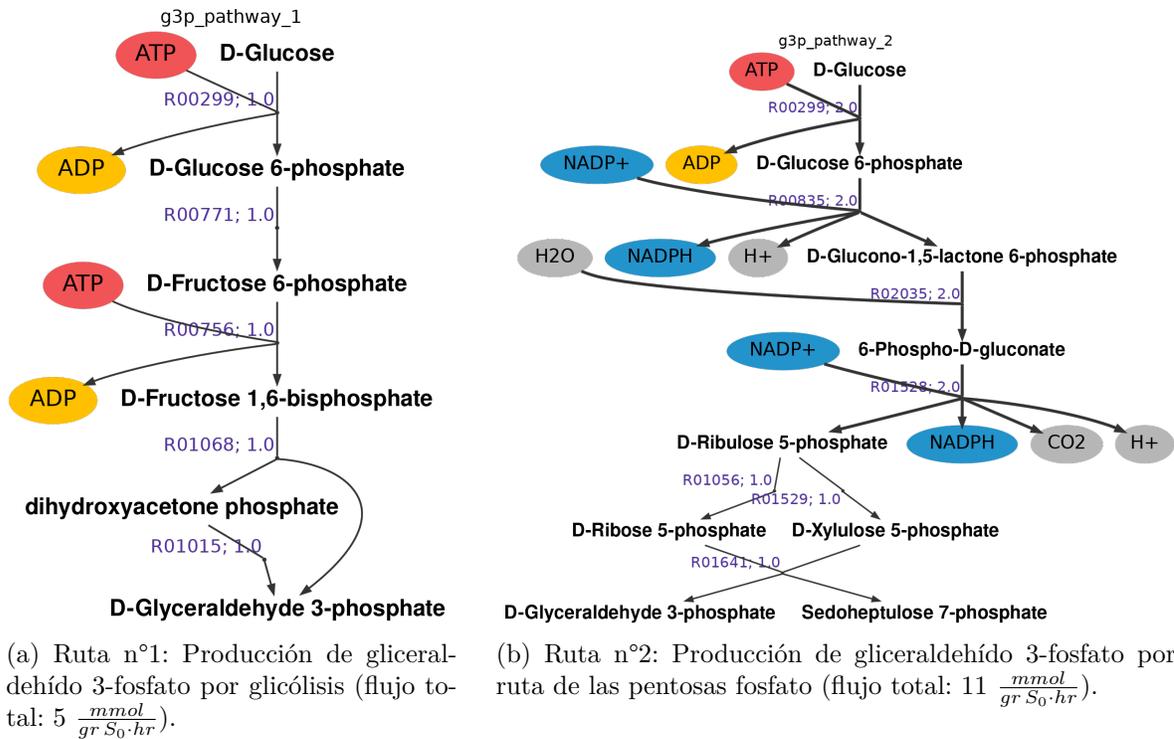


Figura 5.4: Resultados del algoritmo en prueba de producción de gliceraldehído 3-fosfato

Tabla 5.6: Cargas enzimáticas para la producción de gliceraldehído a partir de glucosa

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr.S_0}$)
Ruta 1	R00299	11.3
	R00756	1.80
	R00771	$1.48 \cdot 10^{-1}$
	R01015	-7.19
	R01068	2.19
	Carga total	8.25
Ruta 2	R00299	22.7
	R00835	3.71
	R01056	$-8.34 \cdot 10^{-1}$
	R01528	5.51
	R01529	1.95
	R02035	$3.78 \cdot 10^{-1}$
	Carga total	33.41

La ruta óptima con respecto a flujos para este caso de estudio es vía glicólisis, este resultado es esperable ya que contiene menos reacciones, involucra menos metabolitos que transportan electrones o generan subproductos (H_2O y CO_2) y tiene un mayor rendimiento. La glicólisis genera dos moles de gliceraldehído 3-fosfato por mol de glucosa, en comparación a la segunda ruta donde se requieren dos moles de glucosa para producir un mol del producto deseado.

Por otro lado, analizando los resultados al minimizar carga enzimática, tiene sentido que la segunda ruta obtenida sea subóptima en términos de este caso de estudio, ya que se genera un subproducto que carga con un mol de sustrato inicial por la ruta completa. Es decir, para producir un mol del gliceraldehído-3-fosfato, debe utilizarse el doble de glucosa porque un mol de este sustrato se destina a formar sedoheptulosa-7-fosfato, lo que implica duplicar el uso de las enzimas para tratar los dos moles de glucosa aunque no todo sea destinado al producto deseado.

Al observar los valores obtenidos para las cargas enzimáticas, hay una reacción en cada ruta con valor negativo. Sin embargo, ambas corresponden a reacciones reversibles catalizadas por la misma enzima en ambos sentidos. También se destaca que no todas las reacciones presentes en las rutas obtenidas tienen actividad específica y por esto no es posible considerar su carga enzimática, por ejemplo, la reacción R01641 en la segunda ruta. Esto también influye en el desempeño del algoritmo para un caso en que la red metabólica sea mayor, al no tener uno o más valores de actividad específica no se generan las variables de carga enzimática que definen la función objetivo. Esto podría resultar en que se consideren reacciones con carga enzimática teóricamente nula pero que en la realidad tienen una carga asociada que no está siendo considerada por falta de información.

Con este caso se observó que la implementación de *opStoic* utilizada para este trabajo resulta ser poco flexible ya que el primer paso descrito en la Figura 2.6, que buscaba la estequiometría general óptima pero buscando posibles co-reactivos y co-productos, no es

incluido. Por ende, las entradas y salidas deben especificarse con precisión, si los límites impuestos para alguno de los metabolitos involucrados en estas rutas está errado, la solución esperada no se obtiene. Esto puede generar grandes limitaciones al utilizar el algoritmo para diseñar nuevas vías metabólicas de producción, pues es posible que no se conozcan los posibles co-reactantes o co-productos.

5.2.2.2. Producción de piruvato a partir de distintas fuentes

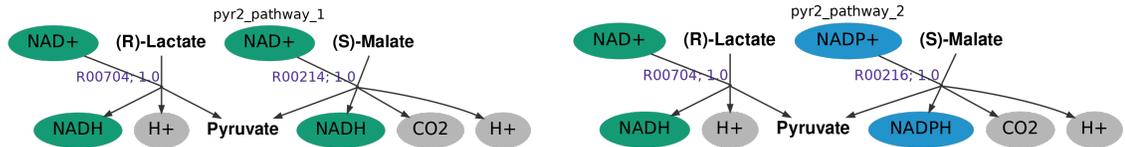
En este caso de estudio se consideraron tres sustratos iniciales posibles: malato, lactato y glucosa. Tanto el malato como el lactato cuentan con dos reacciones directas para formar piruvato, en cambio al utilizar glucosa se espera obtener la ruta de la glicólisis como fue estudiada en el Anexo 8.6.1.

La estequiometría general de este caso consideró límites de entrada y salida que permitieran la obtención de rutas con las tres fuentes de carbono. Se fijó como entrada un mol de cada posible sustrato y como salida entre uno y dos moles de piruvato, ya que el rendimiento es el doble si se utiliza glucosa como fuente de carbono. Adicionalmente, se fijaron posibles entradas y salidas de CO_2 , H_2O , ADP y ATP , iones de hidrógeno y transportadores de electrones. Con esta estequiometría se obtuvieron rutas que solo utilizaban malato o lactato, ya que el algoritmo disminuyó la producción de piruvato para minimizar flujos y carga enzimática. Fue por esto que se decidió imponer la producción de dos moles de piruvato para forzar el uso de glucosa como fuente de carbono, como se observa en la Tabla 5.7. Las rutas obtenidas con el rango original de piruvato se encuentran en el Anexo 8.6.2.

Tabla 5.7: Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de piruvato a partir de distintas fuentes de carbono

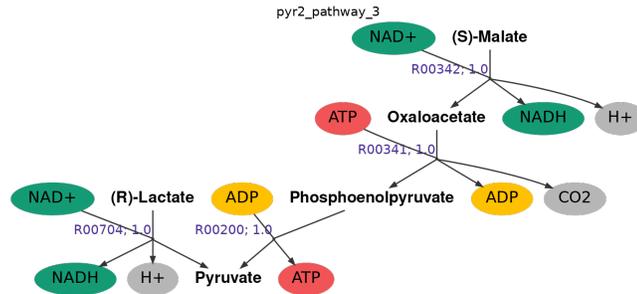
Compuesto	$LB(\frac{mmol}{gr S_0 \cdot hr})$	$UB(\frac{mmol}{gr S_0 \cdot hr})$
Glucosa	-1	0
Malato	-1	0
Lactato	-1	0
Piruvato	2	2
NAD^+	-2	0
$NADH$	0	2
ATP	0	2
ADP	-2	0
$NADP^+$	-2	0
$NAPDH$	0	2
H^+	1	2
H_2O	0	1
CO_2	0	1
Pi	0	2

Con estos límites de entrada y salida se obtienen las rutas de la Figura 5.5.

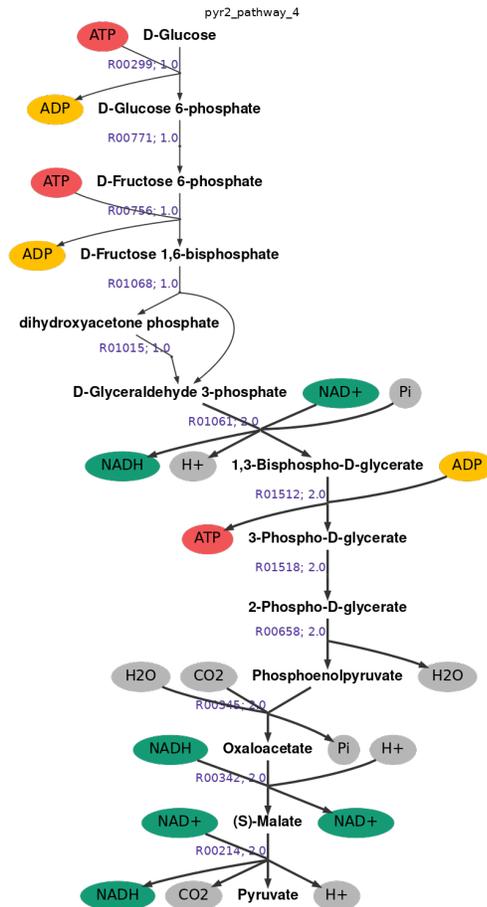


(a) Ruta n°1: producción de piruvato a partir de malato y lactato, reacción directa por $NAD + /NADH$ en ambos casos (flujo total: $2 \frac{mmol}{gr S_0 \cdot hr}$).

(b) Ruta n°2: producción de piruvato a partir de lactato y malato, reacción directa por $NAD + /NADH$ y $NADP + /NADPH$, respectivamente (flujo total: $2 \frac{mmol}{gr S_0 \cdot hr}$).



(c) Ruta n°3: producción de piruvato a partir de lactato y malato por ciclo de Krebs (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).



(d) Ruta n°4: producción de piruvato a partir de glucosa, por ruta de la glicólisis y ciclo de Krebs (flujo total: $19 \frac{mmol}{gr S_0 \cdot hr}$).

Figura 5.5: Resultados del algoritmo en prueba de producción de 2 moles de piruvato minimizando flujos

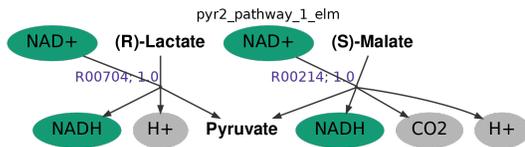
Las primeras tres rutas obtenidas coinciden con las vistas en las soluciones del Anexo 8.6.2, se combinan entre sí para lograr la producción de 2 moles de piruvato. Se habrían obtenido las mismas rutas vistas en la Figura 8.5 con un mayor flujo si se hubiera fijado como entrada entre uno y dos moles para cada fuente de carbono. También se destaca que en la segunda ruta obtenida, se utiliza la reacción directa utilizando malato y $NADP^+/NADPH$ como transportador de electrones, reacción que no fue considerada al fijar la producción entre uno y dos moles de piruvato.

La cuarta ruta obtenida utiliza glucosa como sustrato inicial para producir 2 moles de piruvato en una sola vía sin ramificaciones. Sin embargo, al llegar a fosfoenolpiruvato, que es el último intermediario en la glicólisis, pasa primero por 2 reacciones intermedias antes de llegar a piruvato en lugar de tomar la reacción directa. Las reacciones extra se consideran para consumir y producir otros metabolitos intermedios que se especifican en las entradas y salidas, este es el caso de los iones H^+ , H_2O , grupo fosfato (Pi) y CO_2 . En este caso solo se obtiene H^+ , mientras que los otros 3 compuestos nombrados son producidos y consumidos en igual cantidad. Por sus límites impuestos en la tabla 5.7 no se obtienen como co-productos de salida, si se hubiera aumentado al doble sus límites, estas reacciones extra no habrían sido necesarias. Un ejemplo de esto es el H_2O que tiene como límites entre cero y un mol de salida, es decir, en la solución final del algoritmo se puede producir un mol de agua, se produce y se consume en igual cantidad o bien no se utiliza en absoluto. En la reacción R00658 se producen 2 moles de agua, como este valor queda fuera de los límites establecidos para este compuesto, la reacción R00345 consume el H_2O producido para mantener la estequiometría impuesta. De forma análoga ocurre lo mismo con el grupo fosfato (Pi) en la reacción R00345 y el CO_2 en la reacción R00214.

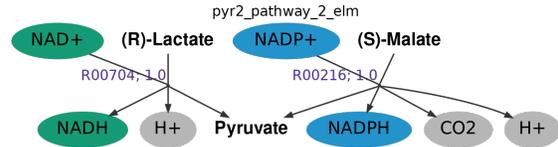
Se destaca también que si existiera una salida de dihidroxiacetona fosfato, podría generarse solo un mol de piruvato a partir de la glicólisis. Después de las reacciones R01068 y R01015 el flujo aumenta al doble al hacer la conversión del metabolito mencionado en el gliceraldehído 3-fosfato y continuar con la segunda parte de esta ruta. Con esto podría haberse obtenido una solución que considerara la glucosa como sustrato inicial para producir un mol de piruvato.

Con todo lo mencionado anteriormente se observa nuevamente que el método *optStoic* es muy estricto con respecto a mantener las entradas y salidas especificadas al no tener integrada la primera parte de su implementación original. En caso de buscar diseñar rutas *de novo* se deben conocer con exactitud los sustratos iniciales, el o los productos y otros metabolitos auxiliares que deben estar presentes. Esto no siempre se cumple, por lo que el uso de la herramienta desarrollada en este trabajo se limita considerablemente. Si se aumentaran los rangos de entrada y salida impuestos es posible que se generen rutas más variadas entre sí, pero que incluyan reacciones cuyo único fin es hacer uso de estos compuestos con rangos más amplios. Por el tamaño de la red metabólica y las rutas consideradas en este caso de estudio, se descarta el ampliar los rangos de entrada y salida en busca de más soluciones. Este enfoque se abordará en el caso de estudio que utiliza la base universal de reacciones.

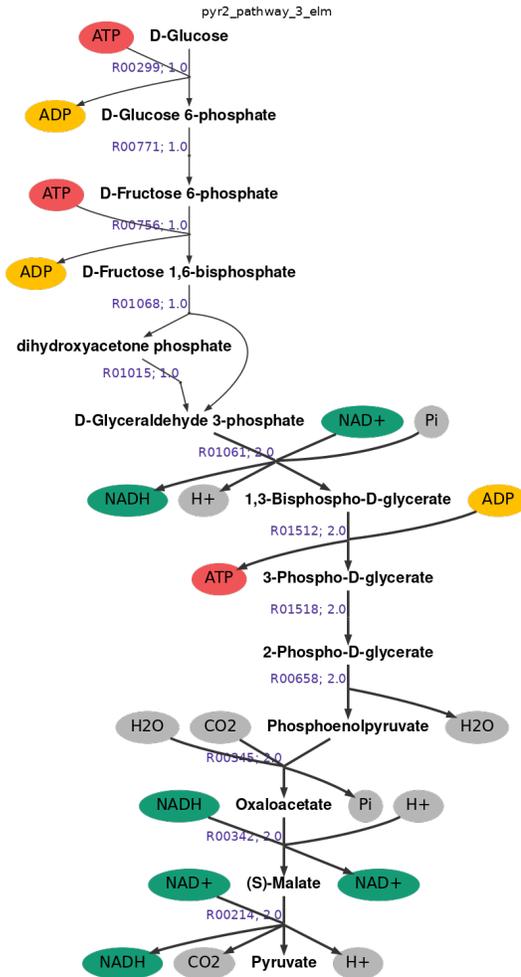
Cambiando el objetivo para minimizar carga enzimática y manteniendo la estequiometría de la Tabla 5.7, se obtienen como resultado las rutas de la Figura 5.6, con sus respectivas cargas enzimáticas en la Tabla 5.8.



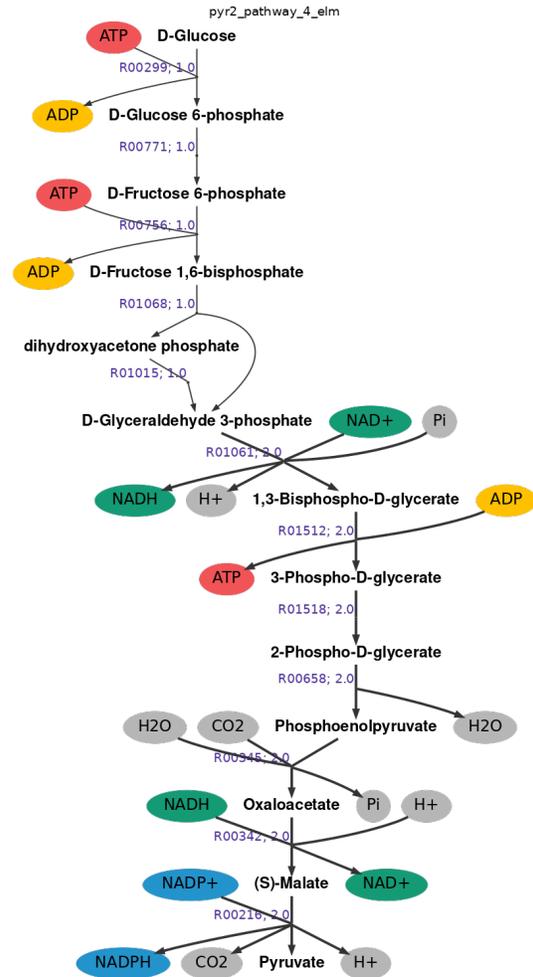
(a) Ruta n°1: producción de piruvato a partir de lactato y malato, reacción directa por $NAD + /NADH$ en ambos casos (flujo total: $2 \frac{mmol}{gr S_0 \cdot hr}$).



(b) Ruta n°2: producción de piruvato a partir de lactato y malato, reacción directa por $NAD + /NADH$ y $NADP + /NADPH$, respectivamente (flujo total: $2 \frac{mmol}{gr S_0 \cdot hr}$).



(c) Ruta n°3: producción de piruvato a partir de glucosa, por glicólisis y ciclo de Krebs (flujo total: $19 \frac{mmol}{gr S_0 \cdot hr}$).



(d) Ruta n°4: producción de piruvato a partir de glucosa, por glicólisis y ciclo de Krebs (flujo total: $19 \frac{mmol}{gr S_0 \cdot hr}$).

Figura 5.6: Resultados del algoritmo en prueba de producción de 2 moles de piruvato minimizando carga enzimática

Tabla 5.8: Cargas enzimáticas para la producción de piruvato a partir de distintas fuentes, límites estrictos

Solución	ID reacción	Carga enzimática ($\frac{mg\text{ enzima}}{gr.S_0}$)
Ruta 1	R00214	5.04
	R00704	4.37
	Carga total	9.41
Ruta 2	R00216	12.6
	R00704	4.37
	Carga total	16.97
Ruta 3	R00214	10.1
	R00299	11.3
	R00342	$-9.36 \cdot 10^{-7}$
	R00345	$-3.78 \cdot 10^{-4}$
	R00658	1.47
	R00756	1.81
	R00771	$1.48 \cdot 10^{-1}$
	R01061	3.15
	R01068	2.19
	R01512	$-1.38 \cdot 10^{-4}$
	Carga total	30.17
Ruta 4	R00216	25.2
	R00299	11.3
	R00342	$-9.36 \cdot 10^{-7}$
	R00345	$-3.78 \cdot 10^{-4}$
	R00658	1.47
	R00756	1.81
	R00771	$1.48 \cdot 10^{-1}$
	R01061	3.15
	R01068	2.19
	R01512	$-1.38 \cdot 10^{-4}$
	Carga total	45.27

Las dos primeras rutas son iguales a las obtenidas al caso análogo de producir dos moles de piruvato minimizando flujos (Figuras 5.5(a) y 5.5(b)). La tercera y la cuarta son análogas a la ruta de la Figura 5.5(d) con la única diferencia de que la última reacción se utiliza $NAD^+/NADH$ o $NADP^+/NADPH$ como transportador de electrones.

En las dos últimas rutas se llega al rendimiento pedido solo a partir de glucosa. Como se mencionó anteriormente, si hubiera una salida de dihidroxiacetona fosfato es probable que el algoritmo hubiera utilizado alguna de las reacciones presentes de las primeras dos rutas de este caso para disminuir el flujo total, ya que luego de la reacción R01068 el flujo aumenta al doble en todos los pasos siguientes.

Los órdenes de magnitud de las cargas enzimáticas para las dos primeras rutas son similares

en todas las reacciones involucradas. En cambio, en las rutas 3 y 4 hay reacciones con signo negativo pero con valores entre cuatro y siete órdenes de magnitud menores que el resto.

Se destaca que las reacciones R00214 y R00216, se diferencian solo en el transportador de electrones que utilizan. Sin embargo, este cambio hace una gran diferencia al minimizar la carga enzimática, pues las rutas que utilizan la reacción R00216 ($NADP^+/NADPH$) tiene una carga neta de más del doble en comparación a las que utilizan R00214 ($NAD^+/NADH$). El uso de una de estas dos reacciones es lo único que diferencia las rutas 1 de la 2 y la 3 de la 4.

En las rutas 3 y 4 hay tres reacciones con carga enzimática neta negativa y de órdenes de magnitud menores a las demás: R00342, R00345 y R01512. La primera reacción es la que tiene menor valor y corresponde a una reacción reversible. La segunda reacción mencionada no es reversible, pero existe una reacción análoga también irreversible que va en el sentido contrario incluyendo el uso de los mismos cofactores. Finalmente, la tercera reacción es la misma que ya se vio en el Anexo 8.6.1 que tiene un comportamiento similar. Las restricciones de carga enzimática resultaron en cargas netas negativas, pero no evitan el flujo por las reacciones que tienen asociadas estas enzimas. Puede decirse que están levemente desplazadas hacia los reactivos.

5.2.3. Base universal de reacciones: producción de isopentenil difosfato (*IPP*)

Debido al interés comercial de este compuesto, se elige como caso de estudio para el universo de reacciones del método *optStoic*. Se buscaron alternativas a la rutas conocidas para la producción de *IPP*. Primero se minimizaron los flujos utilizando el método original y luego la carga enzimática con la información de las actividades específicas.

Inicialmente se consideró glucosa como reactivo inicial y se establecieron como límites de entrada y salida 10 moles para cada compuesto posiblemente involucrado. Sin embargo, este enfoque no obtuvo resultados pues el optimizador alcanzaba el tiempo límite de dos horas sin llegar a una solución. Por esto, se decidió acotar la ruta partiendo directamente desde acetil-CoA en lugar de otras fuentes de carbono. Manteniendo los mismos límites de entrada y salida tampoco se logró obtener un resultado útil, por lo que fue necesario acotar estos límites según la estequiometría propia de la reacción global correspondiente a la ruta del mevalonato (Figura 2.5(A)), esta estequiometría se observa en la Tabla 5.9. Las rutas obtenidas para producir *IPP* desde acetil-CoA con límites de 10 moles de entrada y salida se encuentran en el Anexo 8.6.3.

Tabla 5.9: Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de *IPP* a partir de Acetil-CoA

Compuesto	$LB(\frac{mmol}{gr S_0 \cdot hr})$	$UB(\frac{mmol}{gr S_0 \cdot hr})$
Acetil-CoA	-3	-3
<i>IPP</i>	1	1
<i>NAD</i> ⁺	0	2
<i>NADH</i>	-2	0
<i>NADP</i> ⁺	0	2
<i>NADPH</i>	-2	0
<i>ADP</i>	3	3
<i>ATP</i>	-3	-3
<i>Pi</i>	1	1
<i>H</i> ₂ <i>O</i>	-1	-1
<i>CO</i> ₂	1	1
<i>CoA</i>	3	3
<i>H</i> ⁺	-2	-2

Con estos nuevos límites se lograron obtener soluciones aceptables, tanto al minimizar flujos como al minimizar carga enzimática.

5.2.3.1. Producción de *IPP* minimizando flujos

Desde la Figura 5.7 hasta la 5.11 se pueden apreciar las rutas obtenidas. Todas ellas comparten distintas reacciones, diferenciándose por pasos intermedios o cambiando el transportador de electrones. Debido al parentesco entre algunas soluciones obtenidas por el algoritmo, en el Anexo 8.6.4 se encuentran las rutas que no aparecen en la sección para facilitar la lectura de los resultados.

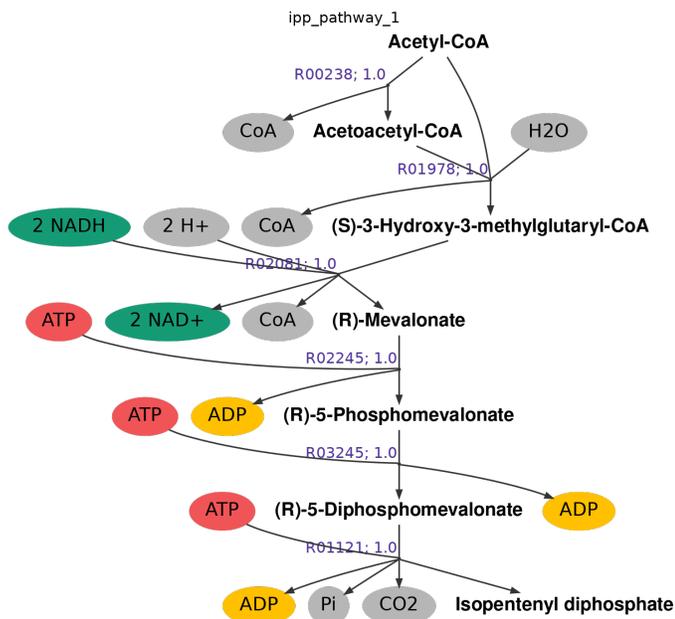


Figura 5.7: Ruta 1: producción de *IPP* a partir de Acetil-coA minimizando flujos. La ruta 2 es análoga, utilizando *NADPH* como transportador de electrones (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).

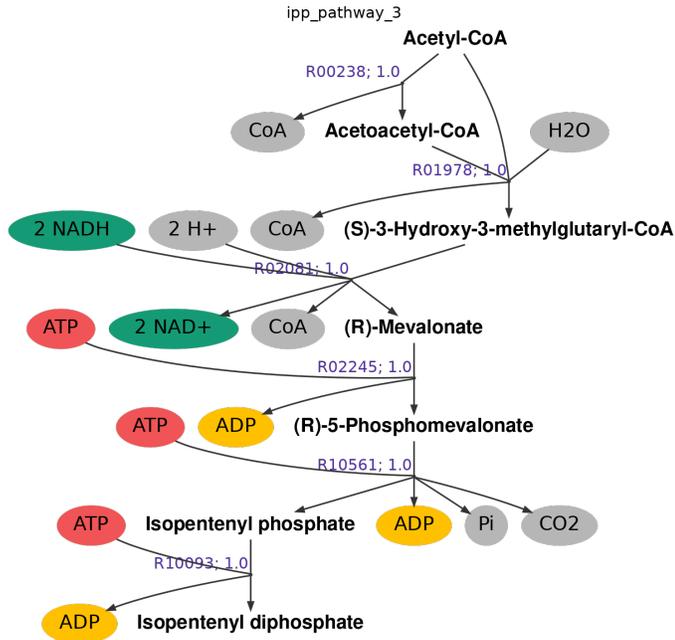


Figura 5.8: Ruta 3: producción de *IPP* a partir de Acetil-coA minimizando flujos. La ruta 4 es análoga, utilizando *NADPH* como transportador de electrones (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).

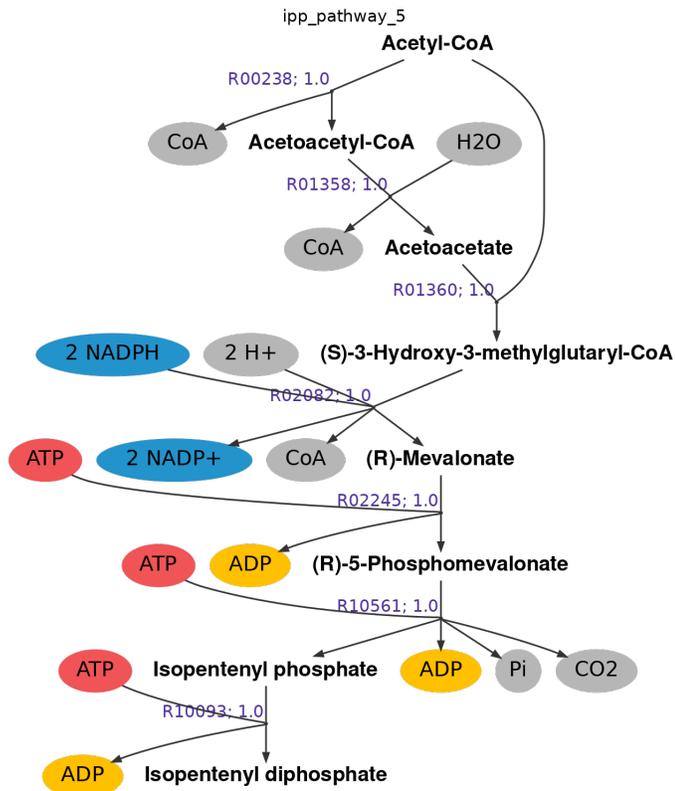


Figura 5.9: Ruta 5: producción de *IPP* a partir de Acetil-coA minimizando flujos. La ruta 7 es análoga, utilizando *NADH* como transportador de electrones (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).

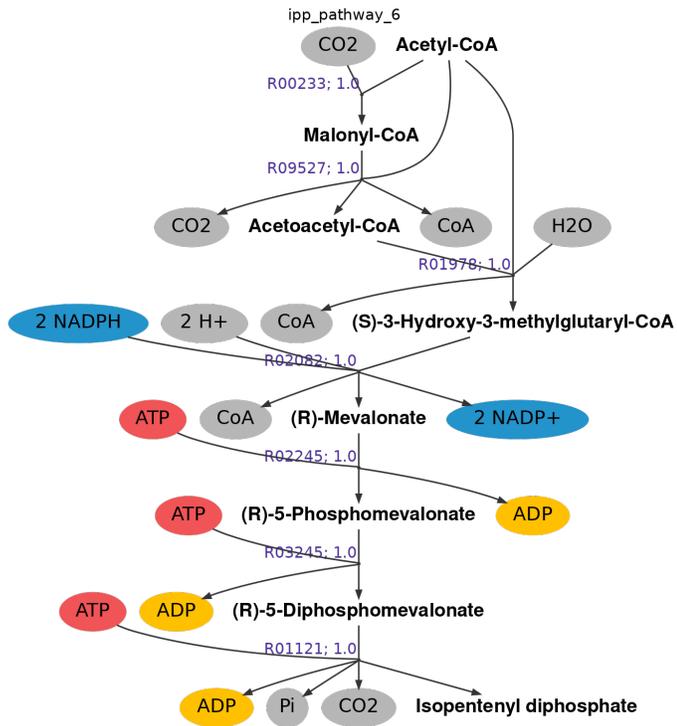


Figura 5.10: Ruta 6: producción de *IPP* a partir de Acetil-coA minimizando flujos. La ruta 10 es análoga, utilizando *NADH* como transportador de electrones (flujo total: $7 \frac{mmol}{gr \cdot 50 \cdot hr}$).

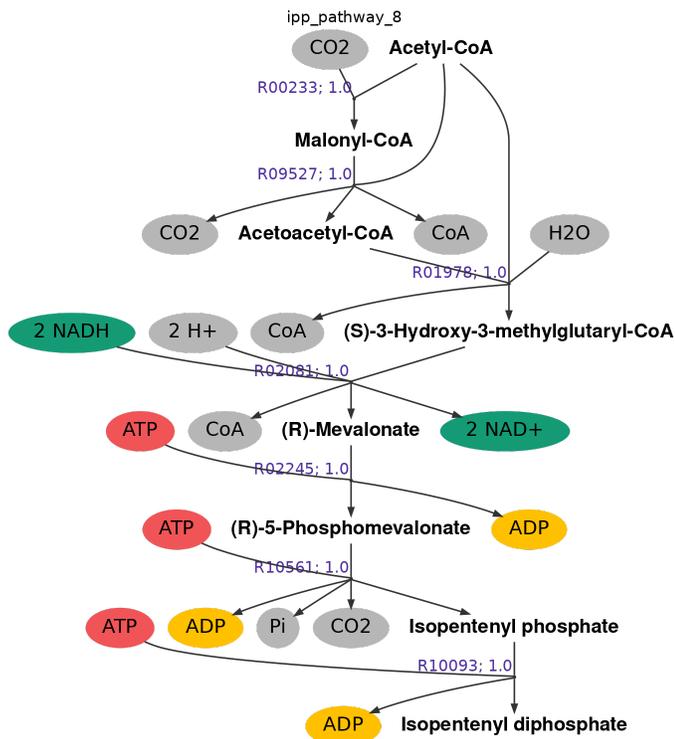


Figura 5.11: Ruta 8: producción de *IPP* a partir de Acetil-coA minimizando flujos. La ruta 9 es análoga, utilizando *NADPH* como transportador de electrones (flujo total: $7 \frac{mmol}{gr \cdot 50 \cdot hr}$).

Las rutas 1 y 2 son equivalentes a la ruta del mevalonato, presente en la Figura 2.5, con la variación de que en la segunda se utiliza *NADPH/NADP⁺* en vez de *NADH/NAD⁺*. Entre las rutas 3 y 4 también se diferencian por el uso de transportador de electrones.

Las rutas 3 y 4 son similares a las dos primeras, pero cambian en el último par de reacciones. En las rutas 1 y 2 se utiliza como último intermediario el 5-difosfomevalonato (R03245) para luego generar *IPP* (R01121), en cambio en las 3 y 4 utilizan isopentenil fosfato (*IP*, por sus siglas en inglés) como intermediario en la reacción R10561 para luego producir *IPP* con la reacción R10093. Es importante notar que al hacer el cambio del par de reacciones R03245-R01121 por el par R10561-R10093 se mantiene la estequiometría de los compuestos *ATP*, *ADP*, *CO₂* y *P_i* que son utilizados en esta sección de las rutas obtenidas. Ambas reacciones de cada par mencionado utilizan *ATP*, pero hay una reacción de cada par donde se producen además *CO₂* y *P_i*.

En las rutas 5 y 7 se agregan las reacciones R01358 y R01360 en reemplazo de la reacción R01978, que pasa de acetoacetyl-CoA a *HMG-CoA* como en las primeras cuatro rutas. En este caso, la ruta 5 utiliza *NADPH/NADP⁺* y la ruta 7 *NADH/NAD⁺*. Pese a que en las rutas mostradas anteriormente se consideraron las rutas más óptimas con *NADH/NAD⁺*, esta elección es irrelevante debido a que el objetivo estaba fijado en la minimización de flujos y en este caso ambas opciones conllevan el mismo flujo.

Las rutas 6, 8, 9 y 10 presentan un resultado similar al discutido con las rutas 5 y 7:

cambian la reacción R00238 que va de acetyl-CoA a acetoacetyl-CoA por el par de reacciones R00233-R09527, que genera primero malonyl-CoA para luego producir acetoacetyl-CoA. En este par de reacciones se consume y se produce CO_2 en iguales cantidades, manteniendo la estequiometría establecida para este compuesto.

Las rutas 6 y 10 continúan después del paso mencionado de forma análoga a la ruta del mevalonato, se diferencian solo en el uso de $NADPH/NADP^+$ y $NADH/NAD^+$, respectivamente, pero utilizan el difosfomevalonato como último intermediario. De forma análoga, las rutas 8 y 9 comparten el *IP* como último intermediario, cumpliendo con las mismas reacciones y estequiometrías de las rutas 3 y 4 para el último par de reacciones, pero se diferencian entre sí por el transportador de electrones.

A modo general, las rutas obtenidas presentan pequeñas variaciones entre ellas y todas son similares a la ruta del mevalonato presente en la literatura [14]. La poca diversidad de las soluciones se debe a la estequiometría general impuesta. Sin embargo, al ampliar los límites de esta estequiometría para obtener mayor diversidad se obtenían rutas demasiado extensas: se incluían las reacciones obtenidas en esta sección pero esas soluciones incorporaban otras reacciones con el único fin de generar y consumir los demás compuestos que poseían mayor holgura (ver Anexo 8.6.3). Esto es una gran limitante para el método desarrollado, pues al no conocer en detalle la estequiometría de la reacción general y sus posibles co-reactivos y co-productos las soluciones no se alejan de la ruta que ya es conocida.

5.2.3.2. Producción de *IPP* minimizando carga enzimática

Desde la Figura 5.12 hasta la 5.16 se pueden apreciar las rutas obtenidas, mientras que en las Tablas 5.10 a la 5.14, aparecen sus respectivas cargas enzimáticas.

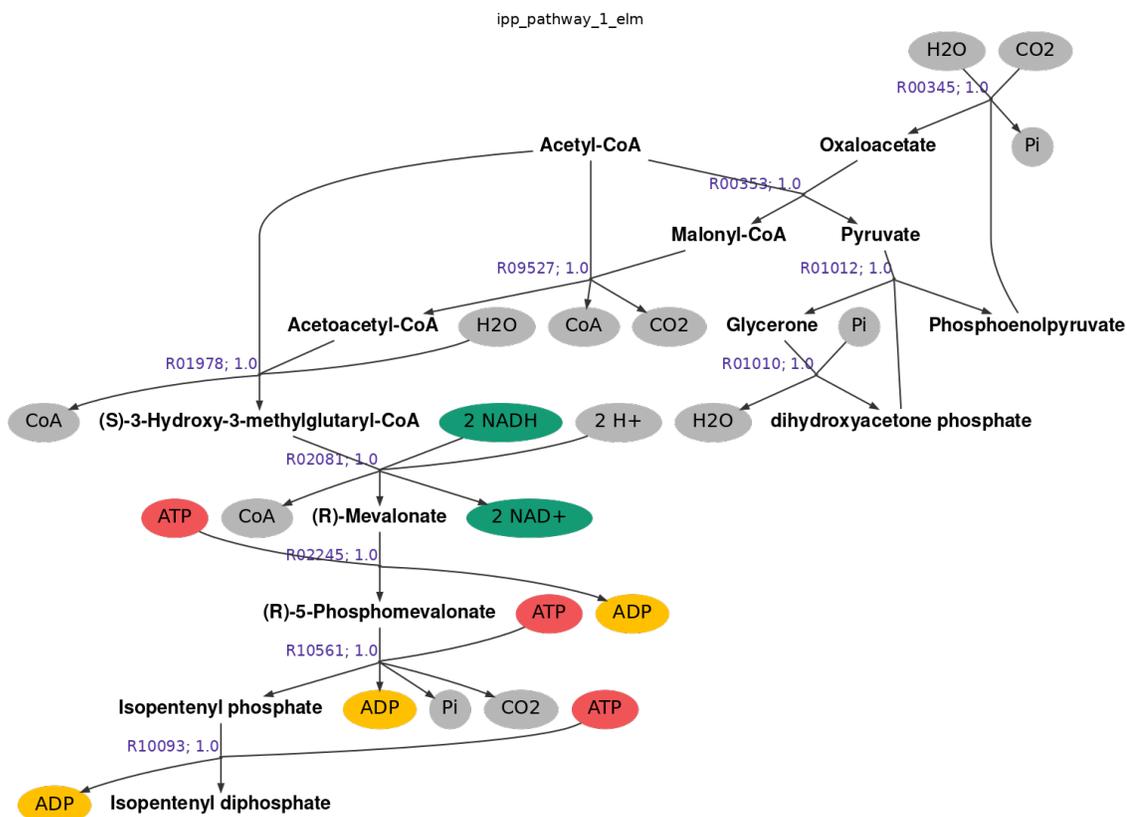


Figura 5.12: Ruta 1: producción de *IPP* a partir de Acetil-coA minimizando carga enzimática (flujo total: $10 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 5.10: Cargas enzimáticas para la producción de *IPP* a partir de acetil CoA (Ruta 1)

Solución	ID reacción	Carga enzimática ($\frac{mg \text{ enzima}}{gr S_0}$)
Ruta 1	R00345	$-1.90 \cdot 10^{-4}$
	R00353	-2.57
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.96
	R02245	4.19
	R10561	11.45
	Carga total	12.10

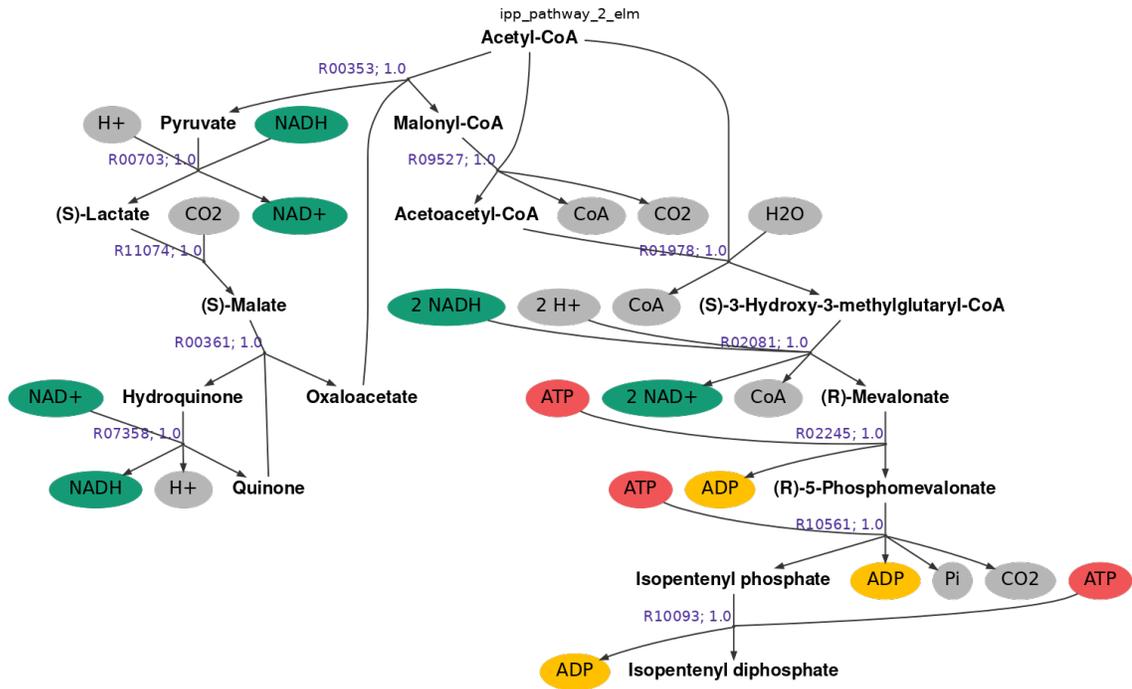


Figura 5.13: Ruta 2: producción de *IPP* a partir de Acetil-coA minimizando carga enzimática (flujo total: $11 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 5.11: Cargas enzimáticas para la producción de *IPP* a partir de acetil CoA (Ruta 2)

Solución	ID reacción	Carga enzimática ($\frac{mg \text{ enzima}}{gr S_0}$)
Ruta 2	R00353	-2.57
	R00703	$-6.32 \cdot 10^{-5}$
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.96
	R02245	4.19
	R10561	11.45
	R11074	-0.14
	Carga total	11.96

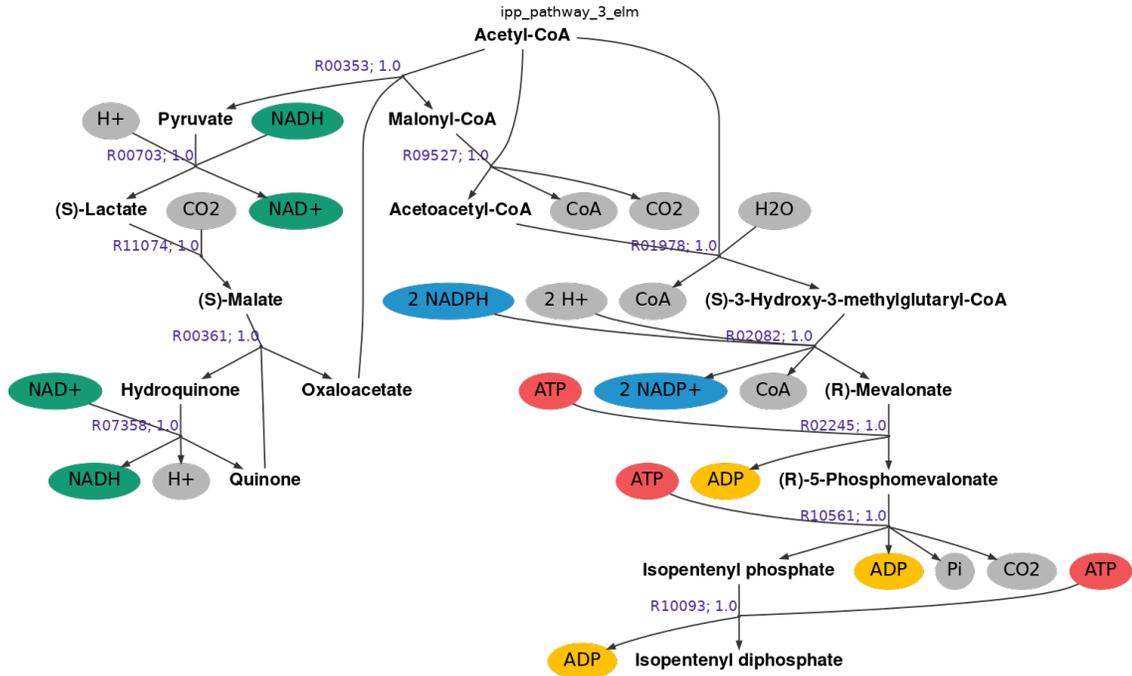


Figura 5.14: Ruta 3: producción de *IPP* a partir de Acetil-coA minimizando carga enzimática (flujo total: $11 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 5.12: Cargas enzimáticas para la producción de *IPP* a partir de acetil CoA (Ruta 3)

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr S_0}$)
Ruta 3	R00353	-2.57
	R00703	$-6.32 \cdot 10^{-5}$
	R01978	$-6.48 \cdot 10^{-3}$
	R02082	-1.57
	R02245	4.19
	R10561	11.45
	R11074	0.14
	Carga total	11.63

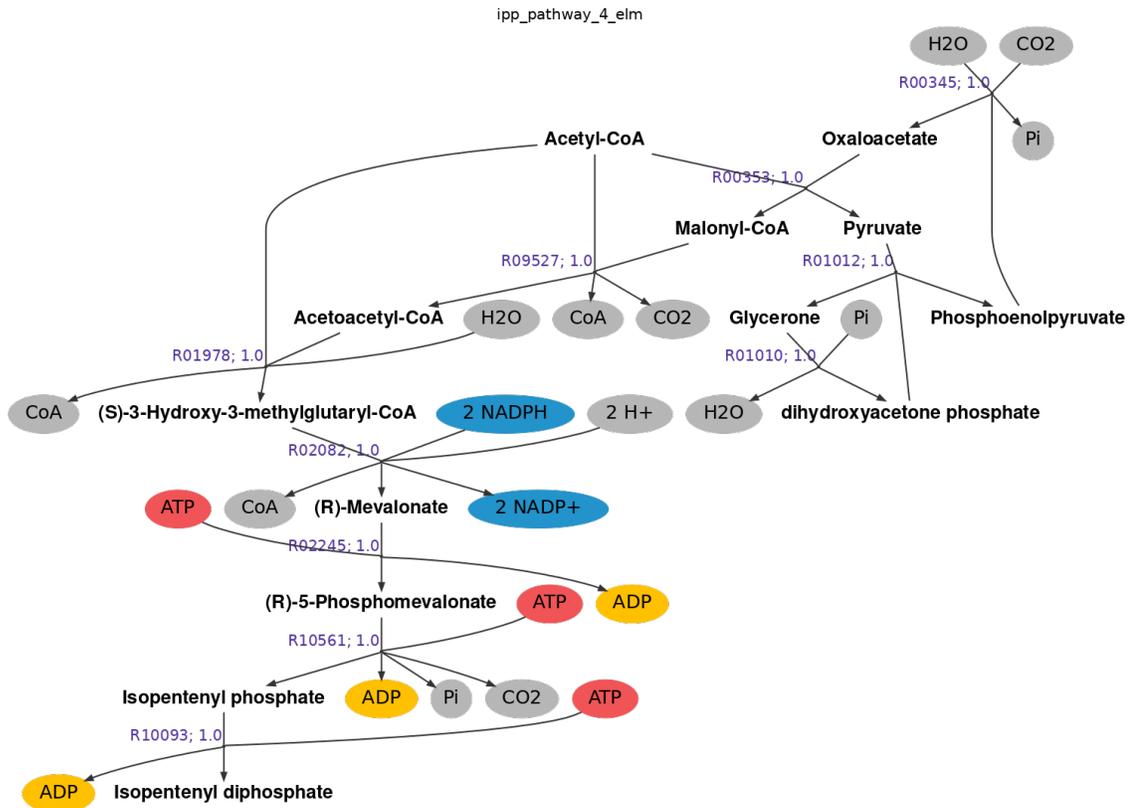


Figura 5.15: Ruta 4: producción de *IPP* a partir de Acetil-coA minimizando carga enzimática (flujo total: $10 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 5.13: Cargas enzimáticas para la producción de *IPP* a partir de acetil CoA (Ruta 4)

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr S_0}$)
Ruta 4	R00345	$-1.90 \cdot 10^{-4}$
	R00353	-2.57
	R01978	$-6.48 \cdot 10^{-3}$
	R02082	-1.57
	R02245	4.19
	R100561	11.45
	Carga total	11.49

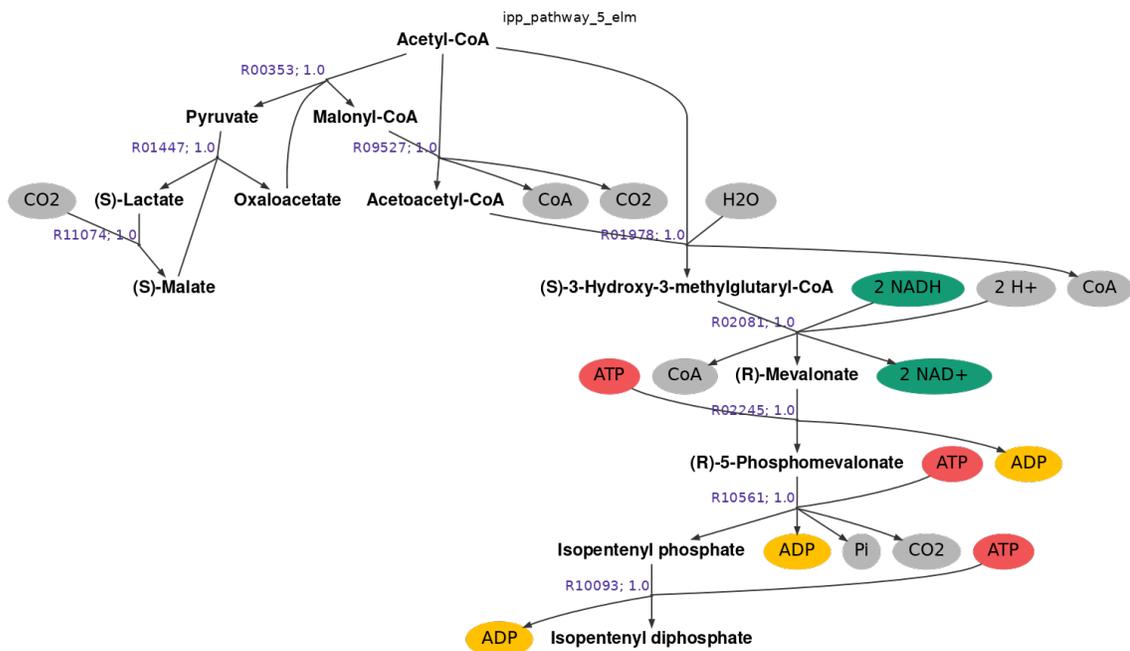


Figura 5.16: Ruta 5: producción de *IPP* a partir de Acetil-coA minimizando carga enzimática (flujo total: $9 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 5.14: Cargas enzimáticas para la producción de *IPP* a partir de acetil CoA (Ruta 5)

Solución	ID reacción	Carga enzimática ($\frac{mg \text{ enzima}}{gr S_0}$)
Ruta 5	R00353	-2.57
	R01447	-0.23
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.95
	R02245	4.19
	R10561	11.45
	R11074	-0.14
	Carga total	11.74

Todas las rutas resultantes al minimizar carga enzimática consideran como últimos pasos desde fosfomevalonato a *IP* (R10561) y luego de éste a *IPP* (R10093). Nunca se llegó a la solución obtenida al minimizar flujos que, en lugar de utilizar *IP*, se utilizaba difosfomevalonato (R03245) y de este compuesto se llegaba al producto final (R01121). Esto se debe a que la reacción R10093 no posee dato de actividad específica, por lo que no conlleva carga enzimática al plantear el problema de optimización y se vuelve más favorable utilizar el par de reacciones R10561-R10093, en teoría.

En cada una de las rutas resultantes no se pasa directamente de acetil-CoA a acetoacetyl-CoA (R00238), sino que toman el camino intermedio por malonil-CoA vista al minimizar flujos en las últimas rutas obtenidas. Esto se debe a que la reacción R00353, asociada a este paso por malonil-CoA, no posee carga enzimática debido a que no se tiene información sobre

su actividad específica, es el mismo comportamiento visto en el uso de *IP* como intermediario. Por ello, para el optimizador representa un camino menos costoso que por la reacción R01978, para la cual si se tiene información de actividad específica y se llega a acetoacetyl-CoA en una reacción pero conlleva una carga mayor.

Otras reacciones presentes en estas rutas se encuentran sin carga enzimática neta por la misma razón, como R00361 en la ruta 3, R07358 en las rutas 2 y 3, R01010 y R01012 en las rutas 1 y 4, R09527 y R10093 en todas las rutas. Esto impacta negativamente los resultados obtenidos, pues se llega a un resultado que es plausible pero se llega a él por falta de información y no porque sea efectivamente la ruta óptima según su cinética. Además, ya se cuenta con la limitante de la estequiometría impuesta en principio, debe ser estricta para evitar que el problema de optimización diverja, por lo que las soluciones obtenidas son similares entre sí.

El uso de transportadores de electrones varía solo en las reacciones R02081 y R02082, utilizando $NAD^+/NADH$ o $NADP^+/NADPH$, respectivamente. Se puede apreciar en las distintas Tablas que la primera reacción posee una menor carga enzimática neta que la segunda, aunque ambas coinciden en tener signo negativo, es decir, que ambas reacciones se encuentran desplazadas hacia los reactivos.

Otro punto interesante es que, si bien faltan valores de actividad específica y la carga total es positiva, las cargas enzimáticas netas que fueron calculadas son en su mayoría negativas. Aunque las reacciones R00345, R00703 y R01978 tienen cargas netas con órdenes de magnitud menores, en general se concluye que todas las rutas se encuentran más desplazadas hacia los reactivos que a los productos en sus primeras reacciones, haciéndolas menos favorables. A partir de esto se infiere que para lograr la producción de *IPP* deben utilizarse mecanismos de inducción para las enzimas encargadas de estas reacciones y así aumentar el rendimiento.

La mayoría de las variaciones entre las soluciones obtenidas ocurre al principio de la ruta. Como la ruta no presenta mayores variaciones desde el mevalonato en adelante, se llevó a cabo un nuevo caso de estudio para evaluar rutas alternativas entre acetyl-CoA y mevalonato [14]. Es en esta sección de la ruta donde se han hecho esfuerzos por encontrar alternativas que sean más favorables termodinámicamente en comparación a la ruta original, con la salvedad de que este paso requiera un gasto de energía adicional [16]. Sin embargo, las rutas generadas por el algoritmo fueron las mismas que se obtuvieron para la producción de *IPP*. El detalle de este caso de estudio adicional se encuentra en el Anexo 8.6.5.

Como fue comentado en los casos de estudio anteriores, la rigidez de la estequiometría no permite explorar otras alternativas que presenten mayores diferencias entre sí. También se buscó obtener soluciones que implicaran un gasto energético en forma de *ATP* para la formación de acetoacetyl-CoA, dado que se ha estudiado de forma experimental [16]. Sin embargo, ninguna de las soluciones obtenidas utiliza este metabolito, el algoritmo tampoco entregó soluciones cuando intentó forzarse su uso.

5.3. Discusiones generales

Ante la promesa de la producción recombinante para distintos compuestos de interés en la industria, se vuelve clave el diseño de rutas metabólicas más eficientes. Para el diseño de estas se han planteado dos enfoques principales: el diseño por medio de herramientas computacionales y el diseño desarrollado experimentalmente. La información disponible en las bases de datos biológicas sigue aumentando, el poder incorporar a los métodos de diseño de rutas esta información les da un mayor sentido biológico a sus resultados.

De las herramientas de diseño *in silico*, *OptStrain* se basa en la presencia o ausencia de ciertas reacciones con respecto al hospedero utilizado [25], *MEMO* se basa en la termodinámica de las reacciones [7] y *optStoic* se basa en la estequiometría de las reacciones y el flujo total de la ruta diseñada [27]. Con el fin de evaluar rutas metabólicas también se han planteado otros métodos. *MDF* permite evaluar qué ruta es más favorable según la termodinámica de sus reacciones [23], mientras que *ECM* se basa en la cantidad de enzima requerida para diferenciar una ruta de otra [24].

Por otro lado, en el diseño de rutas para isoprenoides en específico, se han hecho esfuerzos por reforzar las enzimas clave de la ruta del mevalonato pero esto ha demostrado comprometer el crecimiento del organismo productor [17]. También se logró diseñar e implementar rutas *de novo* a partir de la fosforilación repetitiva de alcoholes C_{5n} , pero se encontraron ciertos cuellos de botella al buscar producir una mayor diversidad de isoprenoides [13].

El método *optStoic* utiliza la estequiometría de las reacciones para buscar rutas fijando como factor decisivo que la suma total de los flujos sea mínima, este enfoque supone que las células utilizan la menor cantidad de recursos para realizar sus funciones. Pero ante distintos escenarios es posible que distintos compuestos se destinen a ciertas rutas por sobre otras para hacer frente a la condición externa en la que se encuentran y esto no puede inferirse a partir de la estequiometría. También existen otros factores que influyen en el flujo de las reacciones como la temperatura, el pH del medio o la eficiencia de las enzimas para catalizar una reacción. En este trabajo se buscó representar todas estas variables al plantear la carga enzimática en función de las actividades específicas de las enzimas. Se buscó predecir rutas metabólicas de producción para el caso del *IPP* con un método basado en restricciones que contempla la estequiometría de las reacciones, pero que también contenga parámetros propios de las enzimas involucradas.

Durante el desarrollo de este trabajo se encontraron distintas dificultades para lograr los objetivos propuestos. En principio, para la obtención de las actividades específicas de la base de datos **BRENDA** fue necesario utilizar su versión descargable, un archivo de texto extenso e irregular, dificultando la apropiada extracción de parámetros. Luego de obtenerlos a partir de su *EC number*, asociarlos a cada reacción según los identificadores de otras bases de datos fue otro desafío. Para el modelamiento realizado con los *GSM* de **BiGG** fue necesario usar como intermediario los identificadores de la base **MetanetX**, que busca unificar las distintas bases de datos biológicas además de realizar análisis de modelos metabólicos [37].

Luego de elegir *optStoic* como base para este proyecto, apareció la limitante de que la

implementación del primer problema de optimización no fue realizada para su versión en `python`, teniendo solo la minimización de flujos para trabajar. Esto conllevó a que fue necesario expresar de forma estricta las entradas y salidas de la ruta que se buscaba diseñar, limitando la exploración de posibles co-reactivos o co-productos desconocidos. También fue necesario volver a realizar el paso intermedio por los identificadores de `MetanetX` ya que el universo de reacciones de `optStoic` utiliza los identificadores de `KEGG`.

Al momento de poner a prueba el método, fue necesario filtrar las actividades específicas con valores de órdenes de magnitud muy grandes ($> 10^9$) o muy pequeños ($< 10^{-9}$), ya que el optimizador no lograba converger por estas diferencias de órdenes de magnitud. Para el caso de estudio de *IPP* también fue necesario acotar la ruta a obtener, empezando desde un metabolito intermedio en lugar de un sustrato presente en el medio de cultivo, ya que no se llegaba a ninguna solución en el tiempo límite de dos horas por la cantidad de reacciones involucradas.

También al analizar los resultados obtenidos, las reacciones que no tienen actividad específica conocida y que aparecen en los resultados distorsionan la interpretación biológica que se buscaba en un principio. Al no tener el parámetro requerido, no se considera carga enzimática para esa reacción y el optimizador la utiliza para obtener una solución, pero en la realidad dicha reacción conlleva una carga enzimática, solo que no está siendo cuantificada. Esto hace que el trabajo realizado pierda funcionalidad por la falta de información para estas reacciones sin actividad específica.

Podría evaluarse el fijar un valor máximo o mínimo para este conjunto de enzimas y así considerar al menos un valor límite de carga enzimática para estos casos. Sin embargo, es posible que se vuelvan a obtener las mismas soluciones vistas en este trabajo, considerando una carga enzimática dada por los límites fijados, pero sin obtener nuevas soluciones debido a la diferencia de órdenes de magnitud entre los valores reales y el límite escogido. Esto se deduce a partir de lo observado en el caso de producción de piruvato (Anexo 8.6.2). Para esta caso se tiene información de las enzimas involucradas y los órdenes de magnitud entre actividades específicas son similares, pero al fijar entre uno y dos moles la producción de piruvato no se utiliza glucosa ya que se da por terminada la optimización antes de obtener la glicólisis como solución. Esto también ocurre al minimizar flujos, aunque la producción de piruvato vía glicólisis es posible, no se obtiene esta solución por la estequiometría impuesta y el largo de la ruta mencionada.

A partir de lo obtenido en la sección 5.2, se puede apreciar que el algoritmo desarrollado se encuentra funcional y reconoce las rutas esperadas entre dos compuestos especificados, a la vez que puede reconocer alternativas que sean similares a la ruta utilizada como base para definir la estequiometría general. Podría utilizarse con vías metabólicas truncadas, tal como se hizo en los últimos casos de estudio, o también en el diseño de rutas que no involucren metabolitos que sean puntos de control de diversas rutas de síntesis en el metabolismo central. Pero no genera alternativas más diversas entre sí por esta necesidad de especificar la estequiometría general en un principio. En caso de conocerse los reactivos y productos en su totalidad, el algoritmo desarrollado puede ser de utilidad ampliando paulatinamente los límites la estequiometría conocida.

Si se consideran límites holgados, las rutas obtenidas contienen reacciones adicionales que solo cumplen la función de utilizar los compuestos de entrada o salida. Si por otro lado, se consideran límites muy estrictos, no se obtienen resultados o se obtienen solo las rutas ya conocidas en las que se basaron estos límites. También se pierde la capacidad de explorar el uso de co-reactivos o la presencia de co-productos, no se obtendrán a menos que sean explicitados en la estequiometría general. Otra cosa que no considera el método desarrollado es la factibilidad de las rutas obtenidas. Si bien el orden en que entrega las soluciones es de menor a mayor carga enzimática total, el fundamento cinético detrás de usar la actividad específica en las restricciones no garantiza que las soluciones también se encuentren en orden de factibilidad termodinámica.

Como propuestas de trabajo futuro se consideran distintos aspectos. Primero, realizar una revisión adicional para asignar un valor de actividad específica para las reacciones que no tienen esta información, ya sea buscarlas de forma manual, asignar un valor predefinido a la actividad específica, cambiar el método de aproximación o utilizar otro parámetro como la constante catalítica incluyendo la conversión de unidades correspondiente. Segundo, implementar en `python` el problema de optimización asociado a obtener la estequiometría óptima e incorporarlo como paso previo a la minimización de la carga enzimática, tal y como fue pensado *optStoic* en su primera implementación. Tercero, aumentar la capacidad de cómputo para poder obtener rutas más grandes en relación a su sustrato inicial y su producto final. Cuarto, ordenar según su factibilidad termodinámica los resultados obtenidos utilizando, por ejemplo, el método *MDF*. Y quinto, realizar pruebas experimentales para poner a prueba las rutas obtenidas por el algoritmo, incorporando las reacciones heterólogas necesarias con técnicas de biología molecular a un organismo hospedero, inducir la producción, medir los rendimientos alcanzados y comparar con la producción de las rutas ya utilizadas actualmente para producir determinado compuesto.

Capítulo 6

Conclusiones

Como primera parte del trabajo, se realizó una extracción de constantes cinéticas y actividades específicas exitosa, incluyendo los parámetros de temperatura y pH necesarios para realizar el filtro. Posteriormente se calculó la constante asociada a la reacción inversa a partir de la obtenida de BRENDA y sus ΔG_{rxn} . Luego, se modeló la carga enzimática modificando un *FBA* al incorporar restricciones de flujo que utilizaran los parámetros extraídos y calculados junto con variables de decisión que representaran esta carga. Se decidió continuar con el uso de las actividades específicas por sobre las constantes catalíticas para evitar utilizar moles de enzima en las unidades y conversiones posteriores, prefiriendo trabajar con gramos de enzima directamente y no necesitar los pesos moleculares de cada enzima.

Como validación de este modelo se utilizaron flujos medidos experimentalmente de distintos cultivos de *E. coli*, donde cada muestra contaba con un gen silenciado. Se compararon los flujos experimentales con los obtenidos por el modelo con el *GSM E. coli core*, al cual se le incorporaron estas restricciones modificadas y se silenciaron los genes respectivos. Al realizar la simulación, si un gen era el único asociado a una reacción y éste era silenciado, dicha reacción no presentaba flujo, mientras que en los datos del cultivo bajaba drásticamente en comparación al caso control. Si el gen silenciado codificaba una proteína que era reemplazable por otra, por la reacción respectiva había un flujo resultante en ambos casos, en los datos experimentales había una disminución con respecto al caso control, mientras que en la simulación se mantenía el mismo flujo que en el caso control, ya que el óptimo se alcanzaba de todas formas. Los casos estudiados consideraban el *knock out* de un único gen por caso, a futuro podrían realizarse simulaciones más complejas al considerar múltiples *knock outs* o utilizar *GSMs* de otros organismos.

La implementación disponible de *optStoic* sirvió como punto de partida para el desarrollo del algoritmo, tanto por el método en sí como por la base de reacciones depurada. Sin embargo, la parte del método asociada a obtener la estequiometría general óptima y sus posibles co-sustratos o co-productos no estaba desarrollada, dejando solo la minimización de flujos. Por esto fue necesario en cada caso de estudio especificar de forma estricta las entradas y salidas de la ruta requerida, limitando la diversidad de soluciones a obtener. Además, solo 2805 de las 6359 reacciones cuentan con actividad específica, lo que afecta el desempeño del algoritmo al omitir restricciones para estas reacciones sin parámetro conocido.

Para los tres universos de reacciones utilizados se obtuvo resultados. El primer conjunto con reacciones ficticias corroboró el funcionamiento del algoritmo en sí. El segundo conjunto obtuvo las rutas esperadas del metabolismo central utilizando actividades específicas reales. Al utilizar el tercer conjunto se obtuvieron rutas similares entre sí para la producción de *IPP*, pero con la falencia de seleccionar reacciones para las cuales no se contaba con actividad específica. Este comportamiento se repitió al explorar otras alternativas en la primera sección de las rutas solución ya obtenidas, las rutas generadas para producir mevalonato fueron las mismas obtenidas al fijar el *IPP* como producto final. Además, para obtener estos resultados se tuvo que utilizar como sustrato inicial acetil-CoA en lugar de glucosa para acortar la ruta y limitar la estequiometría ya que el optimizador no lograba converger en el tiempo límite fijado. Estos cambios limitaron las soluciones a pequeñas variaciones de la ruta del mevalonato presente en la literatura y en la cual se basó la estequiometría impuesta.

Para mejorar las limitaciones descritas anteriormente quedan propuestos distintos trabajos a futuro: asignar un parámetro aproximado u obtenido manualmente para las reacciones que no cuentan con información de la actividad específica; implementar la obtención de la estequiometría óptima presente en la primera implementación de *optStoic*; aumentar la capacidad de cómputo y el tiempo límite para explorar rutas más complejas; evaluar la factibilidad termodinámica de las soluciones obtenidas por el algoritmo y probar en cultivo la producción de *IPP* a partir de las rutas obtenidas que sean, en teoría, más favorables.

Con respecto a los objetivos, se concluye el cumplimiento de todos los objetivos específicos, pero un cumplimiento parcial del objetivo principal, debido a las limitaciones ya comentadas. No obstante, con este trabajo se logra una buena base sobre la cual seguir desarrollando y perfeccionando el algoritmo, presentando también los pasos necesarios para lograr a futuro diseñar rutas de producción de forma más robusta.

Capítulo 7

Nomenclatura

Símbolo	Significado	Unidades
$E_{f,j}$:	Carga enzimática en el sentido directo de la reacción j	$\frac{gr E}{gr S_o}$
$E_{b,j}$:	Carga enzimática en el sentido inverso de la reacción j	$\frac{gr E}{gr S_o}$
E_j :	Carga enzimática neta de la reacción j	$\frac{gr E}{gr S_o}$
<i>FBA</i> :	Análisis de balance de flujos (<i>Flux Balance Analysis</i>)	-
<i>GSM</i> :	Modelo a escala genómica (<i>Genome Scale Model</i>)	-
J^+	Flujo de una reacción en el sentido directo	-
J^-	Flujo de una reacción en el sentido inverso	-
K_j^+ :	Actividad específica en el sentido directo de la reacción j	$\frac{mmolj}{gr E \cdot min}$
K_j^- :	Actividad específica en el sentido inverso de la reacción j	$\frac{mmolj}{gr E \cdot min}$
$K_{eq,j}$:	Constante de equilibrio de la reacción j	-
LB_j :	Límite inferior de la reacción j (<i>Lower Bound</i>)	$\frac{mmolj}{gr S_o \cdot hr}$
<i>LP</i> :	Programación lineal (<i>Linear Programming</i>)	-
<i>MILP</i> :	Programación mixta (<i>Mixed Integer Linear Programming</i>)	-
R :	Constante de los gases ideales	$\frac{J}{mol \cdot K}$
S :	Matriz estequiométrica	-
S_o	Sustrato inicial de una ruta metabólica	gr
T :	Temperatura	K
UB_j :	Límite superior de la reacción j (<i>Upper Bound</i>)	$\frac{mmolj}{gr S_o \cdot hr}$
$v_{f,j}$:	Flujo en el sentido directo de la reacción j	$\frac{mmolj}{gr S_o \cdot hr}$
$v_{b,j}$:	Flujo en el sentido inverso de la reacción j	$\frac{mmolj}{gr S_o \cdot hr}$
v_j :	Flujo neto de la reacción j	$\frac{mmolj}{gr S_o \cdot hr}$
$y_{f,j}$:	Switch de <i>on/off</i> para la reacción j en el sentido directo, toma el valor 1 cuando hay flujo en la reacción en sentido directo o 0 en caso contrario.	-
$y_{b,j}$:	Switch de <i>on/off</i> para la reacción j en el sentido inverso, toma el valor 1 cuando hay flujo en la reacción en sentido inverso o 0 en caso contrario	-
Z :	Función objetivo	-
$\Delta G_{rxn,j}$:	Cambio de energía libre de la reacción j	J

Bibliografía

- [1] B. B. Misra, C. Langefeld, M. Olivier, and L. A. Cox, “Integrated omics: tools, advances and future approaches,” *Journal of molecular endocrinology*, vol. 62, no. 1, pp. R21–R45, 2019.
- [2] P. A. Saa, M. P. Cortés, J. López, D. Bustos, A. Maass, and E. Agosin, “Expanding metabolic capabilities using novel pathway designs: computational tools and case studies,” *Biotechnology journal*, vol. 14, no. 9, p. 1800734, 2019.
- [3] E. J. O’Brien, J. M. Monk, and B. O. Palsson, “Using genome-scale models to predict biological capabilities,” *Cell*, vol. 161, no. 5, pp. 971–987, 2015.
- [4] B. J. Sánchez and J. Nielsen, “Genome scale models of yeast: towards standardized evaluation and consistent omic integration,” *Integrative Biology*, vol. 7, no. 8, pp. 846–858, 2015.
- [5] J. D. Orth, I. Thiele, and B. Ø. Palsson, “What is flux balance analysis?,” *Nature biotechnology*, vol. 28, no. 3, pp. 245–248, 2010.
- [6] A. P. Burgard, P. Pharkya, and C. D. Maranas, “Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization,” *Biotechnology and bioengineering*, vol. 84, no. 6, pp. 647–657, 2003.
- [7] A. v. Kamp and S. Klamt, “MEMO: A method for computing metabolic modules for cell-free production systems,” *ACS synthetic biology*, vol. 9, no. 3, pp. 556–566, 2020.
- [8] J. Nielsen and J. D. Keasling, “Engineering cellular metabolism,” *Cell*, vol. 164, no. 6, pp. 1185–1197, 2016.
- [9] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, “BiGG models: A platform for integrating, standardizing and sharing genome-scale models,” *Nucleic acids research*, vol. 44, no. D1, pp. D515–D522, 2016.
- [10] S. Dev, “Terpenoids,” in *Natural products of woody plants*, pp. 691–807, Springer, 1989.
- [11] B. M. Lange, R. E. Ketchum, and R. B. Croteau, “Isoprenoid biosynthesis. metabolite profiling of peppermint oil gland secretory cells and application to herbicide target analysis,” *Plant Physiology*, vol. 127, no. 1, pp. 305–314, 2001.
- [12] S. C. Kampranis and A. M. Makris, “Developing a yeast cell factory for the production of terpenoids,” *Computational and structural biotechnology journal*, vol. 3, no. 4, p. e201210006, 2012.
- [13] J. M. Clomburg, S. Qian, Z. Tan, S. Cheong, and R. Gonzalez, “The isoprenoid alcohol pathway, a synthetic route for isoprenoid biosynthesis,” *Proceedings of the National Aca-*

- demy of Sciences*, vol. 116, no. 26, pp. 12810–12815, 2019.
- [14] B. M. Lange, T. Rujan, W. Martin, and R. Croteau, “Isoprenoid biosynthesis: the evolution of two ancient and distinct pathways across genomes,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 24, pp. 13172–13177, 2000.
- [15] K. Kemper, M. Hirte, M. Reinbold, M. Fuchs, and T. Brück, “Opportunities and challenges for the sustainable production of structurally complex diterpenoids in recombinant microbial systems,” *Beilstein journal of organic chemistry*, vol. 13, no. 1, pp. 845–854, 2017.
- [16] S. Tippmann, R. Ferreira, V. Siewers, J. Nielsen, and Y. Chen, “Effects of acetoacetyl-coa synthase expression on production of farnesene in *saccharomyces cerevisiae*,” *Journal of Industrial Microbiology and Biotechnology*, vol. 44, no. 6, pp. 911–922, 2017.
- [17] G.-Y. Wang and J. D. Keasling, “Amplification of hmg-coa reductase production enhances carotenoid accumulation in *neurospora crassa*,” *Metabolic engineering*, vol. 4, no. 3, pp. 193–201, 2002.
- [18] K. J. D., N. J. D., and P. D. J., “Method for enhancing production of isoprenoid compounds,” Feb. 27 2007. US Patent 7,183,089.
- [19] L. Heirendt, S. Arreckx, T. Pfau, S. N. Mendoza, A. Richelle, A. Heinken, H. S. Haraldsdóttir, J. Wachowiak, S. M. Keating, V. Vlasov, *et al.*, “Creation and analysis of biochemical constraint-based models using the COBRA toolbox v. 3.0,” *Nature protocols*, vol. 14, no. 3, pp. 639–702, 2019.
- [20] N. E. Lewis, K. K. Hixson, T. M. Conrad, J. A. Lerman, P. Charusanti, A. D. Polpitiya, J. N. Adkins, G. Schramm, S. O. Purvine, D. Lopez-Ferrer, *et al.*, “Omic data from evolved *E. coli* are consistent with computed optimal growth from genome-scale models,” *Molecular systems biology*, vol. 6, no. 1, p. 390, 2010.
- [21] T. Shlomi, O. Berkman, and E. Ruppin, “Regulatory on/off minimization of metabolic flux changes after genetic perturbations,” *Proceedings of the national academy of sciences*, vol. 102, no. 21, pp. 7695–7700, 2005.
- [22] M. R. Antoniewicz, “Dynamic metabolic flux analysis—tools for probing transient states of metabolic networks,” *Current opinion in biotechnology*, vol. 24, no. 6, pp. 973–978, 2013.
- [23] E. Noor, A. Bar-Even, A. Flamholz, E. Reznik, W. Liebermeister, and R. Milo, “Pathway thermodynamics highlights kinetic obstacles in central metabolism,” *PLoS Comput Biol*, vol. 10, no. 2, p. e1003483, 2014.
- [24] E. Noor, A. Flamholz, A. Bar-Even, D. Davidi, R. Milo, and W. Liebermeister, “The protein cost of metabolic fluxes: prediction from enzymatic rate laws and cost minimization,” *PLoS computational biology*, vol. 12, no. 11, p. e1005167, 2016.
- [25] P. Pharkya, A. P. Burgard, and C. D. Maranas, “Optstrain: a computational framework for redesign of microbial production systems,” *Genome research*, vol. 14, no. 11, pp. 2367–2376, 2004.
- [26] A. Chowdhury and C. D. Maranas, “Designing overall stoichiometric conversions and intervening metabolic reactions,” *Scientific reports*, vol. 5, no. 1, pp. 1–20, 2015.

- [27] C. Y. Ng, L. Wang, A. Chowdhury, and C. D. Maranas, “Pareto optimality explanation of the glycolytic alternatives in nature,” *Scientific reports*, vol. 9, no. 1, pp. 1–15, 2019.
- [28] L. Liu, T. P. Cash, R. G. Jones, B. Keith, C. B. Thompson, and M. C. Simon, “Hypoxia-induced energy stress regulates mrna translation and cell growth,” *Molecular cell*, vol. 21, no. 4, pp. 521–531, 2006.
- [29] G. Potters, T. P. Pasternak, Y. Guisez, and M. A. Jansen, “Different stresses, similar morphogenic responses: integrating a plethora of pathways,” *Plant, cell & environment*, vol. 32, no. 2, pp. 158–169, 2009.
- [30] G.-H. Huh, B. Damsz, T. K. Matsumoto, M. P. Reddy, A. M. Rus, J. I. Ibeas, M. L. Narasimhan, R. A. Bressan, and P. M. Hasegawa, “Salt causes ion disequilibrium-induced programmed cell death in yeast and plants,” *The Plant Journal*, vol. 29, no. 5, pp. 649–659, 2002.
- [31] D. Byrne, A. Dumitriu, and D. Segrè, “Comparative multi-goal tradeoffs in systems engineering of microbial metabolism,” *BMC systems biology*, vol. 6, no. 1, pp. 1–19, 2012.
- [32] N. Ishii, K. Nakahigashi, T. Baba, M. Robert, T. Soga, A. Kanai, T. Hirasawa, M. Naba, K. Hirai, A. Hoque, *et al.*, “Multiple high-throughput analyses monitor the response of e. coli to perturbations,” *Science*, vol. 316, no. 5824, pp. 593–597, 2007.
- [33] Y.-Q. Qiu, *KEGG Pathway Database*, pp. 1068–1069. New York, NY: Springer New York, 2013.
- [34] M. Scheer, A. Grote, A. Chang, I. Schomburg, C. Munaretto, M. Rother, C. Söhngen, M. Stelzer, J. Thiele, and D. Schomburg, “Brenda, the enzyme information system in 2011,” *Nucleic acids research*, vol. 39, no. suppl_1, pp. D670–D676, 2010.
- [35] J. O. Maloney, *Perry Chemical Engineers Handbook*. The McGraw-Hill Companies, Inc, 2008.
- [36] A. Flamholz, E. Noor, A. Bar-Even, and R. Milo, “equilibrator—the biochemical thermodynamics calculator,” *Nucleic acids research*, vol. 40, no. D1, pp. D770–D775, 2012.
- [37] S. Moretti, O. Martin, T. Van Du Tran, A. Bridge, A. Morgat, and M. Pagni, “Metanetx/mnxref—reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks,” *Nucleic acids research*, vol. 44, no. D1, pp. D523–D526, 2016.

Capítulo 8

Anexos

8.1. Obtención de parámetros (BRENDA)

En la presente sección se muestra el código principal con el cual se realizó a cabo la extracción de la constante catalítica y la actividad específica de la base de datos BRENDA. Además de estos parámetros se obtuvo información adicional como el rango de pH y temperatura de la enzima, su temperatura y pH óptimos, entre otros parámetros. Por la extensión del código se presenta solo la extracción y filtro de las constantes catalíticas y actividades específicas. La implementación completa se encuentra en el *GitHub* asociado a este proyecto, separado de la implementación del algoritmo, el cual puede encontrarse en el siguiente link: https://github.com/CatalinaGuerraC/Scrapen_BRENDA.

Las enzimas se clasifican en esta base de datos según su *EC number*. Como existen enzimas que cumplen la misma función pero pertenecen a distintos organismos, en la versión descargable se considera un identificador adicional llamado *Protein number* (`pr_number` en el código). Ambos identificadores son utilizados para luego filtrar por temperatura o pH.

Código 8.1: Obtención de la constante catalítica y la actividad específica utilizando expresiones regulares

```
1 def split_pr_numbers(pr_numbers):
2     # brenda_info = [ec_number, pr_numbers, other info (kcat, sa, mw, etc)]
3     splited_pr_numbers = pr_numbers.split(',')
4     return splited_pr_numbers
5
6 def write_info_on_csv(brenda_info, csv_file):
7     r'''
8     Write information extracted from BRENDA_download.txt on a csv file.
9
10    Inputs:
11    -----
12    brenda_info: array with the information to be written per line
13    csv_file: .csv file's name
14    '''
15    for info in brenda_info:
16        ec_number, pr_numbers, *brenda_info_rest = info
17        pr_numbers_list = split_pr_numbers(pr_numbers)
```

```

18     for pr_number in pr_numbers_list:
19         towrite = [ec_number, pr_number, *brenda_info_rest]
20         csv_file.write(SEPARATOR.join(towrite) + '\n')
21
22 def is_mutant(line):
23     'mutant' in line
24
25     ### turnover number (kcat) ###
26
27 def regex_for_kcat(ec_file, ec_number):
28     r'''
29     Obtain the turnover number (kcat) information using regular expressions,
30     it contains the EC number, PR number, kcat value and substrate, it might
31     contain pH and temperature from each reaction, if they're not known appears
32     like 'not specified', if the information corresponds to a mutant strain it
33     will be skipped.
34
35     Inputs:
36     -----
37     ec_file: BRENDA sub file corresponding only to one EC number information
38     ec_number: respective EC number to ec_file
39     Output:
40     -----
41     kcat_values: matrix with kcat information, each row corresponds to a specific
42     enzyme in BRENDA database
43     '''
44     # Pattern looking for: TN #pr_num# Kcat_value {substrate} pH T°
45     kcat_pattern = r'^TN.*#\d*[\d+]*#\ \d+\.\d* \{.*\}.*\[(?P\ \d+\.\d*\)]??.*\[d
↪ +°C]??.*'
46     regex = re.compile(kcat_pattern, flags=re.IGNORECASE|re.MULTILINE)
47     kcat_matches = regex.findall(ec_file)
48     kcat_values = []
49     for kcat_match in kcat_matches:
50         if is_mutant(kcat_match): continue
51         # regular pattern: extract #num# Kcat_value and {substrate}
52         pr_kcat_substrate_pattern = r'^TN.*#(\d*[\d]*)#\ (.*) \{(.*)\}.*'
53         regex2 = re.compile(pr_kcat_substrate_pattern, flags=re.IGNORECASE|re.
↪ MULTILINE)
54         pr_kcat_substrate = regex2.match(kcat_match)
55         # get PR number/kcat value/substrate for every match
56         # pr_numbers could be a list of numbers: #1,2,3,4,5#
57         pr_numbers = pr_kcat_substrate.group(1)
58         kcat_value = pr_kcat_substrate.group(2)
59         substrate = pr_kcat_substrate.group(3)
60         # extract pH value if exists
61         ph_pattern = r'.*pH (\d+\.\d*).*'
62         regex3 = re.compile(ph_pattern, flags=re.IGNORECASE|re.MULTILINE)
63         ph = regex3.match(kcat_match)
64         # get pH value for every match
65         if ph is not None: ph_value = ph.group(1)
66         else: ph_value = 'not specified'
67         # extract T° value if exists

```

```

68     temp_pattern = r'.*(\d\d)°C.*'
69     regex4 = re.compile(temp_pattern, flags=re.IGNORECASE|re.MULTILINE)
70     temp = regex4.match(kcat_match)
71     # get temperature value for every match
72     if temp is not None: temp_value = temp.group(1)
73     else: temp_value = 'not specified'
74     kcat_values.append([ec_number, pr_numbers, kcat_value, substrate, \
75                        ph_value, temp_value])
76     return kcat_values
77
78 def get_kcat(ec_file, ec_number):
79     with open(KCAT_FILE, 'a+') as kcat_file:
80         kcat_info = regex_for_kcat(ec_file, ec_number)
81         write_info_on_csv(kcat_info, kcat_file)
82
83
84 ### specific activity (sa) ###
85
86 def regex_for_sa(ec_file, ec_number):
87     r'''
88     Obtain the specific activity (sa) information using regular expressions,
89     it contains the EC number, PR number and sa value, it might
90     contain pH and temperature from each reaction, if they're not known appears
91     like 'not specified', if the information corresponds to a mutant strain it
92     will be skipped.
93
94     Inputs:
95     -----
96     ec_file: BRENDA sub file corresponding only to one EC number information
97     ec_number: respective EC number to ec_file
98     Output:
99     -----
100    sa_values: matrix with sa information, each row corresponds to a specific
101    enzyme in BRENDA database
102    '''
103    # Pattern looking for: SA #pr_num# sa_value pH T°
104    sa_pattern = r'^SA.*#\d*[\d+]*#\ [0-9]+\.[0-9]*.*[pH \d+\.\d*]?.*[\d+°C]?.*'
105    regex = re.compile(sa_pattern, flags=re.IGNORECASE|re.MULTILINE)
106    sa_matches = regex.findall(ec_file)
107    sa_values = []
108    for sa_match in sa_matches:
109        if is_mutant(sa_match): continue
110        # extract #num# sa_value
111        pr_sa_pattern = r'^SA.*#\d*[\d+]*#\ ([0-9]+\.[0-9]*).*'
112        regex2 = re.compile(pr_sa_pattern, flags=re.IGNORECASE|re.MULTILINE)
113        pr_sa = regex2.match(sa_match)
114        # get PR number/specific activity value for every match
115        # pr_numbers could be a list of numbers: #1,2,3,4,5#
116        pr_numbers = pr_sa.group(1)
117        sa_value = pr_sa.group(2)
118        # extract pH value if exists
119        ph_pattern = r'.*pH (\d+\.\d*).*'

```

```

120     regex3 = re.compile(ph_pattern, flags=re.IGNORECASE|re.MULTILINE)
121     ph = regex3.match(sa_match)
122     # get pH value for every match
123     if ph is not None: ph_value = ph.group(1)
124     else: ph_value = 'not specified'
125     # extract T° value if exists
126     temp_pattern = r'.*(\d\d)°C.*'
127     regex4 = re.compile(temp_pattern, flags=re.IGNORECASE|re.MULTILINE)
128     temp = regex4.match(sa_match)
129     # get temperature value for every match
130     if temp is not None: temp_value = temp.group(1)
131     else: temp_value = 'not specified'
132     sa_values.append([ec_number, pr_numbers, sa_value, ph_value, \
133                     temp_value])
134     return sa_values
135
136 def get_sa(ec_file, ec_number):
137     with open(SA_FILE, 'a+') as sa_file:
138         sa_info = regex_for_sa(ec_file, ec_number)
139         write_info_on_csv(sa_info, sa_file)
140
141

```

Código 8.2: Filtro de la constante catalítica y la actividad específica por temperatura y pH

```

1  ## Filtering by pH: ("not specified" must be included)
2  kcat_with_ph_info = kcat_df.query('pH != "not specified"',inplace = False)
3  #convert str type to float
4  kcat_with_ph_info['pH'] = kcat_with_ph_info['pH'].apply(pd.to_numeric)
5  #pH must be < 8.0 and > 4.0
6  ubandlb_kcat_with_ph_info = kcat_with_ph_info.query('(pH <= 8.0) and (pH >= 4.0)'
7  ↪ \
8  , inplace = False)
9  #include "not specified" info
10 kcat_without_ph_info = kcat_df.query('pH == "not specified"',inplace = False)
11
12 ## concatenate ph filtered kcats
13 ph_filtered_kcats = [ubandlb_kcat_with_ph_info,kcat_without_ph_info]
14 ph_filtered_kcat_df = pd.concat(ph_filtered_kcats,ignore_index=False)
15
16 ## Filtering by T°: ("not specified" must be included)
17 kcat_with_temp_info = ph_filtered_kcat_df.query('Temperature != "not specified"',\
18         inplace = False)
19 kcat_with_temp_info['Temperature'] = kcat_with_temp_info['Temperature'].apply(pd.
20 ↪ to_numeric)
21 ub_kcat_with_temp_info = kcat_with_temp_info.query(('Temperature <= 40'), \
22         inplace = False)
23 #include "not specified" info
24 kcat_without_temp_info = ph_filtered_kcat_df.query('Temperature == "not specified"'
25 ↪ ,\
26         inplace = False)
27
28 ## concatenate temp and ph filtered kcats
29 temp_filtered_kcat = [ub_kcat_with_temp_info,kcat_without_temp_info]
30
31 filtered_kcat_df = pd.concat(temp_filtered_kcat,ignore_index=False)
32
33 ## Filtering by pH: ("not specified" must be included)
34 sa_with_ph_info = sa_df.query('pH != "not specified"',inplace = False)
35 #convert str type to float
36 sa_with_ph_info['pH'] = sa_with_ph_info['pH'].apply(pd.to_numeric)
37 #pH must be < 8.0 and > 4.0
38 ubandlb_sa_with_ph_info = sa_with_ph_info.query('(pH <= 8.0) and (pH >= 4.0)', \
39         inplace = False)
40 sa_without_ph_info = sa_df.query('pH == "not specified"',inplace = False)
41
42 ## concatenate ph filtered kcats
43 ph_filtered_sa = [ubandlb_sa_with_ph_info,sa_without_ph_info]
44 ph_filtered_sa_df = pd.concat(ph_filtered_sa,ignore_index=False)
45
46 ## Filtering by T°: ("not specified" must be included)
47 sa_with_temp_info = ph_filtered_sa_df.query('Temperature != "not specified"',\

```

```
47         inplace = False)
48 sa_with_temp_info['Temperature'] = sa_with_temp_info['Temperature'].apply(pd.
    ↳ to_numeric)
49 ub_sa_with_temp_info = sa_with_temp_info.query(('Temperature <= 40'), \
50         inplace = False)
51 #include "not specified" info
52 sa_without_temp_info = ph_filtered_sa_df.query('Temperature == "not specified"',\
53         inplace = False)
54
55 #concatenate df3 and df4
56 temp_filtered_sa = [ub_sa_with_temp_info,sa_without_temp_info]
57 filtered_sa_df = pd.concat(temp_filtered_sa,ignore_index=True)
58
```

Código 8.3: Cálculo de la actividad específica en el sentido inverso de las reacciones

```
1 universal_db = pd.read_excel('./universal_metabolic_DB_curated_with_brenda_info.  
  ↪ xlsx', index_col=0)  
2  
3 universal_db['Reverse_specific_activity'] = '' * len(universal_db)  
4 universal_db['Kcat_minus'] = '' * len(universal_db)  
5  
6 R = 8.314/1000 #ideal gases constant [kJ/(mol K)]  
7 T = 298 # temperature(?) [K]  
8 for i in universal_db.index:  
9     if not pd.isna(universal_db.at[i,'Specific_activity']) and not \  
10    pd.isna(universal_db.at[i, 'ΔGr (kJ/mol) mean']):  
11         sa_minus = universal_db.at[i,'Specific_activity'] \  
12         /np.exp(-1*(universal_db.at[i, 'ΔGr (kJ/mol) mean']/(R*T)))  
13         universal_db.at[i, 'Reverse_specific_activity'] = sa_minus  
14     if not pd.isna(universal_db.at[i,'Kcat']) and not \  
15    pd.isna(universal_db.at[i, 'ΔGr (kJ/mol) mean']):  
16         kcat_minus = universal_db.at[i,'Kcat'] \  
17         /np.exp(-1*(universal_db.at[i, 'ΔGr (kJ/mol) mean']/(R*T)))  
18         universal_db.at[i, 'Kcat_minus'] = kcat_minus  
19
```

8.2. Modelamiento de la carga enzimática

Código 8.4: Minimización de la carga enzimática para un *GSM*

```
1 def opt_enzymatic_load(model, db, solution=None, frac=1):
2     with model:
3         model.solver = 'glpk'
4         add_opt_enzymatic_load(model,db,solution,frac)
5         new_solution = model.optimize()
6         print("Status:", new_solution)
7         print("Objective value:", model.objective.value)
8         print('\n')
9         for var in model.variables:
10            print(var.name, ":", var.primal)
11         return new_solution
12
13 def add_opt_enzymatic_load(model, db, solution=None, frac=1):
14
15     r"""
16     Optimize ecoli model including energy load restrictions (E_i)
17
18     This function optimize a given model adding variables and constraints
19     that represents the energy load of reactions, which are related with
20     turnover number (Kcat) and changing the original objective to sumatory
21     of energy loads (sum(E_i)). It looks for minimize this sumatory.
22
23     Modified constraints: fluxes - E_i * Kcat- <= v_i <= E_i * Kcat+
24
25     Inputs:
26     -----
27     model : cobra.Model
28         The model to add variables and constraints and objective to.
29     db : pandas Dataframe
30         The database in dataframe format which has the Kcat values for every
31         reaction in the model
32
33     Outputs:
34     -----
35     solution: cobra.Solution
36         The solution for this new optimization problem, including objective_value
37         and fluxes of every reaction
38     """
39     # set biomass flux value to v_opt (default problem)
40     if solution is None:
41         solution = model.slim_optimize()
42     v_biomass = 0.8739215069684305
43     clean_db = db
44     #clean_db = clean_database(model, db)
45     prob = model.problem
46     reactions = model.reactions
47     vars_and_cons = []
```

```

48     obj_vars = []
49     exchanges = model.exchanges
50     # change objective to 0
51     biomass_constraint = prob.Constraint(model.objective.expression, lb=v_biomass, ub=
↪ v_biomass)
52     model.objective = prob.Objective(Zero, direction='min', sloppy=True)
53     vars_and_cons.append(biomass_constraint)
54     for rxn in model.reactions:
55         if (rxn in exchanges): continue
56         # skip reactions without GPR or GPR = s0001 (including non-enzimatic transport
↪ reactions)
57         if rxn.gene_reaction_rule is None or rxn.gene_reaction_rule == 's0001': continue
58         v = rxn.flux_expression
59         if rxn.id in clean_db.index:
60             # change reversibility of irreversible reactions (rxn.rev = False to rxn.rev = True
↪ )
61             if not rxn.reversibility: rxn.lower_bound = -1000
62             # Variable: energy cost of flux_i
63             E = prob.Variable('E_' + rxn.id, lb=0, ub=1000)
64             vars_and_cons.append(E)
65             # v_i <= E_i*K+
66             # v_i - E_i*K+ <= 0
67             sa_plus = clean_db.at[rxn.id, 'Specific_activity']
68             if sa_plus:
69                 UBi = prob.Constraint(v - E * sa_plus, ub=0, name='cons_plus_' + rxn.id)
70                 vars_and_cons.append(UBi)
71             # - E_i*K- <= v_i
72             # 0 <= v_i + E_i*K-
73             sa_minus = clean_db.at[rxn.id, 'Reverse_specific_activity']
74             if sa_minus:
75                 LBi = prob.Constraint(v + E * sa_minus, lb=0, name='cons_minus_' + rxn
↪ .id)
76                 vars_and_cons.append(LBi)
77                 obj_vars.append(E)
78             model.add_cons_vars(vars_and_cons)
79             # Set linear coefficients in objective (sum(Ei))
80             model.objective.set_linear_coefficients({v: 1.0 for v in obj_vars})
81
82             # objective, energy load variables and their constraints
83             print(model.objective, end='\n')
84             print("-----")
85             for const in model.constraints:
86                 print(const, end='\n')
87             print("-----")
88

```

8.3. Simulación de la carga enzimática en un *GSM* y comparación con datos experimentales

Desde las tablas 8.1 hasta la 8.5 se pueden apreciar los valores numéricos de los flujos cuantificados obtenidos de literatura [32] y los flujos simulados con el modelo *E. coli core* minimizando carga enzimática. Las figuras 8.1, 8.2 y 8.3 corresponden a las comparaciones entre los datos experimentales y las simulaciones que no fueron mostradas en la sección de resultados.

Tabla 8.1: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b4025 (proteína PGI)

Flujo estudiado	Datos experimentales		Predicción del modelo	
	Caso control	<i>Knock out</i>	Caso control	<i>Knock out</i>
GLCpts	3,28	2,73	1,51	1,53
PGI	2,85	0	1,36	0
PFK	2,92	0,93	1,28	0,83
FBA	2,92	0,93	1,28	0,83
TPI	2,92	0,93	1,28	0,83
PGK	5,84	3,28	2,46	2,03
PGM	5,54	2,98	2,16	1,73
PYK	1,57	0,14	0,18	0,01

Tabla 8.2: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b2097 (proteína fbaB)

Flujo estudiado	Datos experimentales		Predicción del modelo	
	Caso control	<i>Knock out</i>	Caso control	<i>Knock out</i>
GLCpts	3,28	2,76	1,51	1,51
PGI	2,85	2,01	1,36	1,36
PFK	2,92	2,29	1,28	1,28
FBA	2,92	2,29	1,28	1,28
TPI	2,92	2,29	1,28	1,28
PGK	5,84	4,66	2,46	2,46
PGM	5,54	4,39	2,16	2,16
PYK	1,57	1,24	0,18	0,18

Tabla 8.3: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b1723 (proteína pfkB)

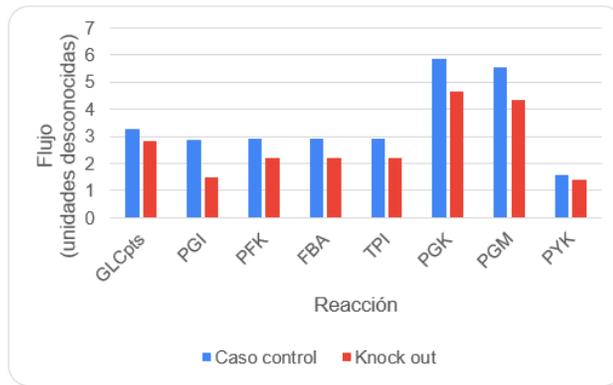
Flujo estudiado	Datos experimentales		Predicción del modelo	
	aso control	<i>Knock out</i>	aso control	<i>Knock out</i>
GLCpts	3,28	2,84	1,51	1,51
PGI	2,85	1,48	1,36	1,36
PFK	2,92	2,19	1,28	1,28
FBA	2,92	2,19	1,28	1,28
TPI	2,92	2,19	1,28	1,28
PGK	5,84	4,63	2,46	2,46
PGM	5,54	4,35	2,16	2,16
PYK	1,57	1,39	0,18	0,18

Tabla 8.4: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b1854 (proteína pykA)

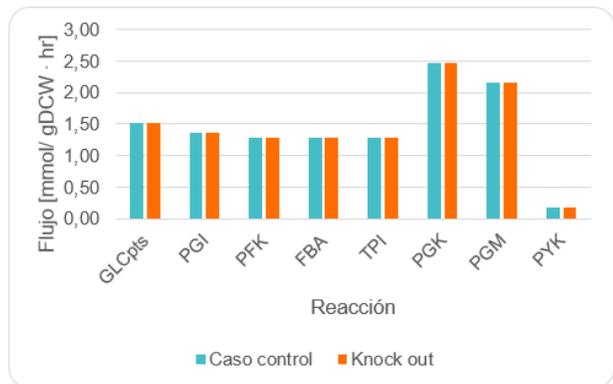
Flujo estudiado	Datos experimentales		Predicción del modelo	
	Caso control	<i>Knock out</i>	Caso control	<i>Knock out</i>
GLCpts	3,28	2,82	1,51	1,51
PGI	2,85	2,09	1,36	1,36
PFK	2,92	2,37	1,28	1,28
FBA	2,92	2,37	1,28	1,28
TPI	2,92	2,37	1,28	1,28
PGK	5,84	4,79	2,46	2,46
PGM	5,54	4,51	2,16	2,16
PYK	1,57	1,44	0,18	0,18

Tabla 8.5: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b1702 (proteína ppsA)

Flujo estudiado	Datos experimentales		Predicción del modelo	
	Caso control	<i>Knock out</i>	Caso control	<i>Knock out</i>
GLCpts	3,28	3,00	1,51	1,51
PGI	2,85	2,04	1,36	1,36
PFK	2,92	2,46	1,28	1,28
FBA	2,92	2,46	1,28	1,28
TPI	2,92	2,46	1,28	1,28
PGK	5,84	5,10	2,46	2,46
PGM	5,54	4,80	2,16	2,16
PYK	1,57	1,05	0,18	0,18

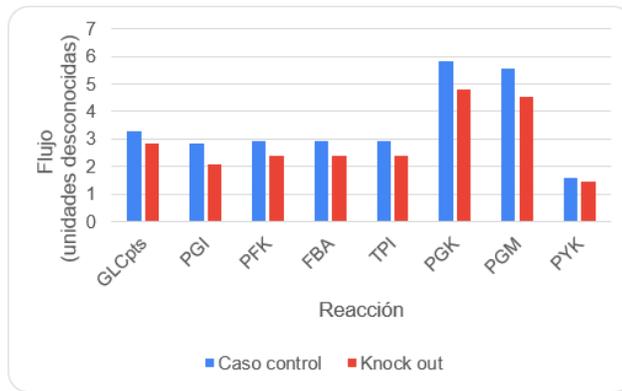


(a) Datos experimentales

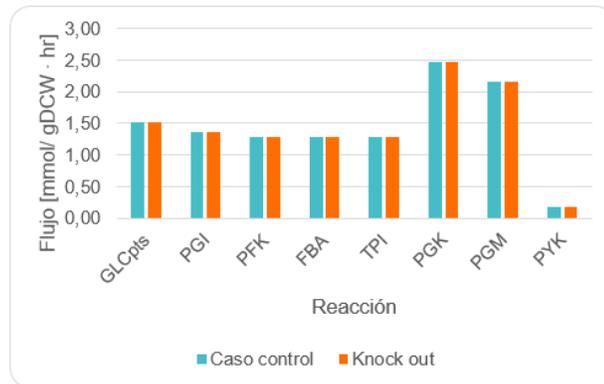


(b) Simulación

Figura 8.1: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b1723 (proteína pfkB)

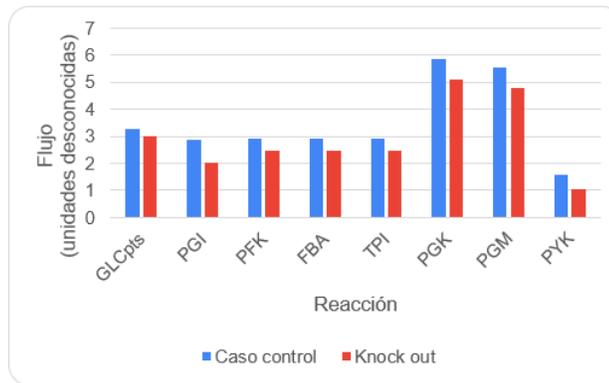


(a) Datos experimentales

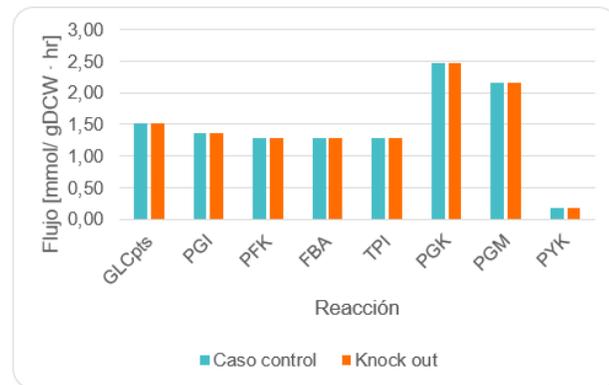


(b) Simulación

Figura 8.2: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b1854 (proteína pykA)



(a) Datos experimentales



(b) Simulación

Figura 8.3: Comparación del modelo con datos experimentales de bibliografía: *knock out* gen b1702 (proteína ppsA)

8.4. Modificaciones a *optStoic*

Dado que el trabajo original está organizado en formato de librería, en esta sección solo se incluye el planteamiento del problema minimización de flujos o carga enzimática y la función que obtiene la solución a dicho problema (`create_minflux_problem(...)` y `solve(...)`, respectivamente).

Las principales diferencias con el trabajo original son la creación de variables de carga enzimática en los sentidos directo e inverso de cada reacción además de las variables ya existentes, la redefinición de la función objetivo como la suma de las variables de carga enzimática, la creación de las restricciones respectivas para los flujos con las actividades específicas y la obtención las cargas netas como salida en las soluciones.

El código completo se encuentra en el repositorio de *GitHub* asociado a este trabajo disponible en https://github.com/CatalinaGuerraC/enzyme_load_minimization.

Código 8.5: Planteamiento del problema de optimización para minimizar flujos o carga enzimática.

```
1 def create_minflux_problem(self):
2     """
3     Create minflux/minRxn LP problem (a pulp LpProblem object)
4     minimize z = sum(j\j_exchange, vf(j) + vb(j))
5     subject to:
6         z = zlb
7         v(j) = vf(j) - vb(j)
8         sum(j, S(i,j) * v(j)) = 0, for all i
9         vf(j) >= yf(j) * eps, for all j
10        vf(j) <= yf(j) * M, for all j
11        vb(j) >= yb(j) * eps, for all j
12        vb(j) <= yb(j) * M, for all j
13        yf(j) + yb(j) <= 1, for all j
14
15        for reaction in jloop (not in jblock)
16            sum(j, Nint(l,j) * G(j)) = 0, for all l
17            G(j) >= -M * a(j) + (1 - a(j))
18            G(j) <= -a(j) + M * (1 - a(j))
19            v(j) >= -M * (1 - a(j))
20            v(j) <= M * a(j)
21
22    """
23    self.logger.info("Formulating problem...")
24    # Scalar
25    M = self.M
26
27    # Initialize variables
28    if self.objective == 'MinFlux':
29        v = pulp.LpVariable.dicts("v", self.database.reactions,
30                                lowBound=-M, upBound=M, cat=self._varCat)
31        vf = pulp.LpVariable.dicts("vf", self.database.reactions,
```

```

32         lowBound=0, upBound=M, cat=self.__varCat)
33     vb = pulp.LpVariable.dicts("vb", self.database.reactions,
34                               lowBound=0, upBound=M, cat=self.__varCat)
35     yf = pulp.LpVariable.dicts("yf", self.database.reactions,
36                               lowBound=0, upBound=1, cat='Binary')
37     yb = pulp.LpVariable.dicts("yb", self.database.reactions,
38                               lowBound=0, upBound=1, cat='Binary')
39
40     if self.objective == 'EnzymeLoad':
41         Ef = pulp.LpVariable.dicts("Ef", self.database.reactions,
42                                   lowBound=0, upBound=1e+10, cat='Continuous')
43         Eb = pulp.LpVariable.dicts("Eb", self.database.reactions,
44                                   lowBound=0, upBound=1e+10, cat='Continuous')
45         E = pulp.LpVariable.dicts("E", self.database.reactions,
46                                   lowBound=-1e10, upBound=1e10, cat='Continuous')
47         v = pulp.LpVariable.dicts("v", self.database.reactions,
48                                   lowBound=-M, upBound=M, cat=self.__varCat)
49         vf = pulp.LpVariable.dicts("vf", self.database.reactions,
50                                   lowBound=0, upBound=M, cat=self.__varCat)
51         vb = pulp.LpVariable.dicts("vb", self.database.reactions,
52                                   lowBound=0, upBound=M, cat=self.__varCat)
53         yf = pulp.LpVariable.dicts("yf", self.database.reactions,
54                                   lowBound=0, upBound=1, cat='Binary')
55         yb = pulp.LpVariable.dicts("yb", self.database.reactions,
56                                   lowBound=0, upBound=1, cat='Binary')
57
58     if self.add_loopless_constraints:
59         a = pulp.LpVariable.dicts("a", self.database.reactions,
60                                   lowBound=0, upBound=1, cat='Binary')
61         G = pulp.LpVariable.dicts("G", self.database.reactions,
62                                   lowBound=-M, upBound=M, cat='Continuous')
63     else:
64         a = None
65         G = None
66
67     # Update lower and upper bound based on reaction directionality
68
69     for j in self.database.reactions:
70
71         if j in self.database.excluded_reactions:
72             v[j].lowBound = 0
73             v[j].upBound = 0
74             vf[j].lowBound = 0
75             vf[j].upBound = 0
76             yf[j].lowBound = 0
77             yf[j].upBound = 0
78             yb[j].lowBound = 0
79             yb[j].upBound = 0
80
81             #continue
82
83     if self.database.rxntype[j] == 0:

```

```

84     # Forward irreversible
85     v[j].lowBound = 0
86     v[j].upBound = M
87     yb[j].upBound = 0
88     vb[j].upBound = 0
89
90     elif self.database.rxntype[j] == 1:
91         # Reversible
92         v[j].lowBound = -M
93         v[j].upBound = M
94
95     elif self.database.rxntype[j] == 2:
96         # Reverse irreversible
97         v[j].lowBound = -M
98         v[j].upBound = 0
99         vf[j].upBound = 0
100        yf[j].upBound = 0
101
102        elif self.database.rxntype[j] == 4:
103            v[j].lowBound = 0
104            v[j].upBound = 0
105
106        # Fix stoichiometry of source/sink metabolites
107        for rxn, bounds in self.specific_bounds.items():
108            v[rxn].lowBound = bounds['LB']
109            v[rxn].upBound = bounds['UB']
110
111        LB = {}
112        UB = {}
113
114        for j in self.database.reactions:
115            LB[j] = v[j].lowBound
116            UB[j] = v[j].upBound
117
118        lp_prob = pulp.LpProblem("OptStoic", pulp.LpMinimize)
119
120        # Min-Rxn objective
121        if self.objective == 'MinRxn':
122            condition = pulp.lpSum([yf[j] + yb[j]
123                                   for j in self.database.reactions
124                                   if self.database.rxntype[j] != 4])
125            lp_prob += condition, "MinRxn"
126
127        # Min-Flux objective
128        elif self.objective == 'MinFlux':
129            condition = pulp.lpSum([vf[j] + vb[j]
130                                   for j in self.database.reactions
131                                   if self.database.rxntype[j] != 4])
132            lp_prob += condition, "MinFlux"
133
134        if self.zlb is not None:
135            # fix lower bound

```

```

136         lp_prob += condition == self.zlb, 'zLowerBound'
137
138     # Enzyme-Load objective
139     elif self.objective == "EnzymeLoad":
140         condition = pulp.lpSum([Ef[j] + Eb[j]
141                                 for j in self.database.reactions
142                                 if self.database.rxntype[j] !=4])
143         lp_prob += condition, "EnzymeLoad"
144
145         if self.zlb is not None:
146             # fix lower bound
147             lp_prob += condition == self.zlb, 'zLowerBound'
148
149     # Constraints
150     # Mass_balance
151     for i in self.database.metabolites:
152         # If metabolites not involve in any reactions
153         if i not in self.database.S:
154             continue
155         label = "mass_balance_%s" % i
156         dot_S_v = pulp.lpSum([self.database.S[i][j] * v[j]
157                               for j in list(self.database.S[i].keys())])
158         condition = dot_S_v == 0
159         lp_prob += condition, label
160
161     # if self.objective == 'MinRxn':
162     # for j in self.database.reactions:
163     #     lp_prob += v[j] >= y[j]*LB[j], "cons1_%s" %j
164     #     lp_prob += v[j] >= y[j]*LB[j], "cons1_%s" %j
165     #     lp_prob += v[j] <= y[j]*UB[j], "cons2_%s" %j
166
167     if self.objective == 'MinFlux':
168         for j in self.database.reactions:
169             lp_prob += (v[j] == vf[j] - vb[j]), "flux_%s" % j
170
171             # These constraints ensure that when yf=0 and yb=0 ,
172             # no flux goes through the reaction
173             lp_prob += vf[j] >= yf[j] * 0.5, "cons1_%s" % j
174             lp_prob += vf[j] <= yf[j] * M, "cons2_%s" % j
175             lp_prob += vb[j] >= yb[j] * 0.5, "cons3_%s" % j
176             lp_prob += vb[j] <= yb[j] * M, "cons4_%s" % j
177             # Ensure that either yf or yb can be 1, not both
178             lp_prob += yf[j] + yb[j] <= 1, 'cons5_%s' % j
179
180     if self.objective == "EnzymeLoad":
181         sa_db = pd.read_excel('./enzyme_load_info.xlsx',
182                               index_col= 'KEGG ID')
183         # replace empty strings with np.NaN
184         sa_db = sa_db.replace(r'\s*$', np.NaN, regex=True)
185         # replace np.NaN with zeros
186         sa_db = sa_db.fillna(0)
187         #divide sa_minus and sa_plus valid coefficients

```

```

188 sa_plus_db = sa_db.query('(sa_plus<=1e9) and (sa_plus>=1e-9)')
189 sa_minus_db = sa_db.query('(sa_minus<=1e9) and (sa_minus>=1e-9)')
190
191 for j in self.database.reactions:
192     lp_prob += (v[j] == vf[j] - vb[j]), "flux_%s" % j
193     lp_prob += (E[j] == Ef[j] - Eb[j]), "enzyme-load_%s" % j
194     # Ensure that either yf or yb can be 1, not both
195     lp_prob += yf[j] + yb[j] <= 1, 'direction_%s' % j
196     # Cases: constraint with and without specific activity
197     # values (sa_plus or sa_minus)
198     if (j in sa_plus_db.index) and (sa_plus_db.at[j,'sa_plus'] != 0):
199         lp_prob += vf[j] <= Ef[j]*60*sa_plus_db.at[j,'sa_plus'], "sa_plus_cons_%s" %
↪ j
200         lp_prob += Ef[j] <= yf[j] * Ef[j].upBound
201         if (j in sa_minus_db.index) and (sa_minus_db.at[j,'sa_minus'] != 0):
202             lp_prob += vb[j] <= Eb[j]*60*sa_minus_db.at[j,'sa_minus'], "
↪ sa_minus_cons_%s" % j
203             lp_prob += Eb[j] <= yb[j] * Eb[j].upBound
204             # These constraints ensure that when yf=0 and yb=0 ,
205             # no flux goes through the reaction
206             lp_prob += vf[j] >= yf[j] * 0.5, "cons1_%s" % j
207             lp_prob += vf[j] <= yf[j] * M, "cons2_%s" % j
208             lp_prob += vb[j] >= yb[j] * 0.5, "cons3_%s" % j
209             lp_prob += vb[j] <= yb[j] * M, "cons4_%s" % j
210
211 if self.add_loopless_constraints:
212     self.logger.info("Loopless constraints are turned on.")
213
214     loop_rxn = list(set(self.database.internal_rxns) -
215                     set(self.database.blocked_rxns))
216
217     # Loopless constraints
218     for l in self.database.loops:
219         label = "loopless_cons_%s" % l
220         dot_N_G = pulp.lpSum([self.database.Ninternal[l][j] * G[j]
221                               for j in list(self.database.Ninternal[l].keys())])
222         condition = dot_N_G == 0
223         lp_prob += condition, label
224
225     for j in loop_rxn:
226         lp_prob += G[j] >= -M * a[j] + (1 - a[j]), "llcons1_%s" % j
227         lp_prob += G[j] <= -a[j] + M * (1 - a[j]), "llcons2_%s" % j
228         lp_prob += v[j] >= -M * (1 - a[j]), "llcons3_%s" % j
229         lp_prob += v[j] <= M * a[j], "llcons4_%s" % j
230
231     # Fix nad(p)h production and consumption
232     if self.custom_flux_constraints is not None:
233         self.logger.info("Adding custom constraints...")
234
235     for group in self.custom_flux_constraints:
236         lp_prob += pulp.lpSum(v[rxn] for rxn in group['reactions']
237                               ) <= group['UB'], "%s_UB" % group['constraint_name']

```

```
238         lp_prob += pulp.lpSum(v[rxn] for rxn in group['reactions']
239                               ) >= group['LB'], "%s_LB" % group['constraint_name']
240
241     if self.objective == 'EnzymeLoad':
242
243         return lp_prob, v, vf, vb, yf, yb, a, G, E, Ef, Eb
244     else:
245         return lp_prob, v, vf, vb, yf, yb, a, G
```

Código 8.6: Función solución al problema de minimización de flujos o carga enzimática.

```
1 def solve(
2     self,
3     exclude_existing_solution=True,
4     outputfile=None,
5     max_iteration=None):
6     """
7     Solve OptStoic problem using pulp.solvers interface
8
9     Args:
10        exclude_existing_solution (bool, optional): If True, create and add integer cut
11            constraints for pathways that are found using the same OptStoic instance,
12            but solved in previous function call.
13        outputfile (str, optional): name of outputfile
14        max_iteration (None, optional): Externally specified maximum number of pathway to
15        ↪ be
16            found using OpStoic. If not specified, it will set to the internal max iterations.
17
18    Returns:
19        TYPE: Description
20
21    Raises:
22        ValueError: Description
23    """
24    if self.objective not in ['MinFlux', 'MinRxn', 'EnzymeLoad']:
25        raise ValueError(
26            "The objective for OptStoic is not correctly defined. Please use either 'MinFlux' or '
27            ↪ MinRxn'.")
28
29    if max_iteration is None:
30        max_iteration = self.max_iteration
31
32    self.logger.info(
33        "Finding multiple pathways using Optstoic %s...",
34        self.objective)
35
36    if self.objective == 'EnzymeLoad':
37
38        lp_prob, v, vf, vb, yf, yb, a, G, E, Ef, Eb = self.create_minflux_problem()
39    else:
40        lp_prob, v, vf, vb, yf, yb, a, G = self.create_minflux_problem()
41
42    # Create integer cut for existing pathways
43    if exclude_existing_solution and bool(self.pathways):
44        self.iteration = max(self.pathways.keys()) + 1
45        if self.iteration > max_iteration:
46            raise ValueError('Max iteration is less than current '
47                'iteration. Increase max_iteration '
48                'before solving!')
```

```

48
49     for ind, pathway in self.pathways.items():
50         rxnlist = list(set(pathway.reaction_ids_no_exchange))
51         condition = pulp.lpSum(
52             [(1 - yf[j] - yb[j]) for j in rxnlist]) >= 1
53         lp_prob += condition, "IntegerCut_ %d" % ind
54
55     self.logger.info("Solving problem...")
56     # if self.iteration == 1:
57     #     result_output = open(os.path.join(self.result_filepath, outputfile), "w+")
58     # else:
59     #     result_output = open(os.path.join(self.result_filepath, outputfile), "a+")
60
61     #enzyme_loads dict as an output
62     enzyme_loads = {}
63
64     while True and self.iteration <= max_iteration:
65         self.logger.info("Iteration %s", self.iteration)
66         # lp_prob.wr("OptStoic.lp", mip=1) # optional
67         e1 = time.time()
68         lp_prob.solve(solver=self.pulp_solver)
69         e2 = time.time()
70         self.logger.info(
71             "This iteration solved in %.3f seconds.",
72             (e2 - e1))
73
74         # The solution is printed if it was deemed "optimal
75         if pulp.LpStatus[lp_prob.status] == "Optimal":
76             self.logger.info("Writing result to output file...")
77             # result_output.write("\nIteration no.: %d\n" %self.iteration)
78             # result_output.write("\nModelstat: %s\n" %pulp.LpStatus[lp_prob.status])
79             res = {}
80             res['reaction_id'] = []
81             res['flux'] = []
82             res['iteration'] = self.iteration
83             res['time'] = (e2 - e1)
84             res['modelstat'] = "Optimal"
85             res['EnzymeLoad'] = []
86
87             # save enzyme loads for each iteration
88             elm_actual_iteration = {}
89             for j in self.database.reactions:
90                 if v[j].varValue is not None:
91                     if v[j].varValue > EPS or v[j].varValue < -EPS:
92                         res['reaction_id'].append(j)
93                         res['flux'].append(v[j].varValue)
94                     if self.objective == 'EnzymeLoad':
95                         elm_actual_iteration['E_' + j] = E[j].varValue
96             enzyme_loads['iteration_' + str(self.iteration)] = elm_actual_iteration
97
98
99     #         result_output.write(" %s %.8f\n" %(v[j].name, v[j].varValue))

```

```

100     # result_output.write("%s = %.8f\n" % (self.objective, pulp.value(lp_prob.
101     ↪ objective)))
102     # result_output.write("-----\n\n")
103
104     integer_cut_reactions = list(
105         set(res['reaction_id']) - set(self.database.user_defined_export_rxns))
106
107     self.pathways[self.iteration] = Pathway(
108         id=self.iteration,
109         name='Pathway_{:03d}'.format(self.iteration),
110         reaction_ids=res['reaction_id'],
111         fluxes=res['flux'],
112         sourceSubstrateID='C00031',
113         endSubstrateID='C00022',
114         note=res
115     )
116     if outputfile == None:
117         outputfile = "OptStoic_pulp_result.txt"
118     self.write_pathways_to_json(json_filename=outputfile)
119
120     # Integer cut constraint is added so that
121     # the same solution cannot be returned again
122     condition = pulp.lpSum([(1 - yf[j] - yb[j])
123                             for j in integer_cut_reactions]) >= 1
124     lp_prob += condition, "IntegerCut_ %d" % self.iteration
125     #print([(1 - yf[j] - yb[j]) for j in integer_cut_reactions])
126     #print(integer_cut_reactions)
127     #print(condition)
128     self.iteration += 1
129
130     # If a new optimal solution cannot be found, end the program
131     else:
132         break
133
134     # result_output.close()
135
136     self.lp_prob = lp_prob
137
138     if self.objective == 'EnzymeLoad':
139         return self.lp_prob, self.pathways, enzyme_loads
140     else:
141         return self.lp_prob, self.pathways

```

8.5. Reacciones consideradas en el metabolismo central

Para evaluar el desempeño del modelo se consideró una red metabólica que involucre las siguientes vías del metabolismo central: glicólisis, ciclo de Krebs y vía de las pentosas fosfato. Para ello se utilizó como base el modelo *E. coli core* de la base de datos BiGG [9], el cual consta de 72 metabolitos, 95 reacciones y 137 genes. El formato de este modelo metabólico fue adaptado a los identificadores de KEGG, no se consideraron las reacciones de transporte con el medio extracelular y se agregó la reacción R00299, que es el primer paso de la glicólisis pero en este modelo se cambia por la reacción GLCpts, la cual es una reacción de transporte donde se fosforila la glucosa del medio externo por intercambio con fosfoenolpiruvato.

ID KEGG	ID BiGG	Ecuación
R00086	ATPM	$ATP + H_2O \Leftrightarrow ADP + \text{Ortofosfato}$
R00112	NADTRHD/ THD2	$NADPH + NAD^+ \Leftrightarrow NADP + NADH$
R00114	GLUSy	$2 \text{ L-Glutamato} + NADP^+ \Leftrightarrow$ $\text{L-Glutamina} + 2\text{-Oxoglutarato} + NADPH + H^+$
R00127	ADK1	$ATP + AMP \Leftrightarrow 2ADP$
R00199	PPS	$ATP + \text{Piruvato} + H_2O \Leftrightarrow$ $AMP + \text{Fosfoenolpiruvato} + \text{Ortofosfato}$
R00200	PYK	$ATP + \text{Piruvato} \Leftrightarrow ADP + \text{Fosfoenolpiruvato}$
R00209	PDH	$\text{Piruvato} + CoA + NAD^+ \Leftrightarrow$ $\text{Acetil-CoA} + CO_2 + NADH + H^+$
R00212	PFL	$\text{Acetil-CoA} + \text{Formato} \Leftrightarrow CoA + \text{Piruvato}$
R00214	ME1	$(S)\text{-Malato} + NAD^+ \Leftrightarrow$ $\text{Piruvato} + CO_2 + NADH + H^+$
R00216	ME2	$(S)\text{-Malato} + NADP^+ \Leftrightarrow$ $\text{Piruvato} + CO_2 + NADPH + H^+$
R00228	ACALD	$\text{Acetaldehído} + CoA + NAD^+ \Leftrightarrow$ $\text{Acetil-CoA} + NADH + H^+$
R00230	PTAr	$\text{Acetil-CoA} + \text{Ortofosfato} \Leftrightarrow$ $CoA + \text{Acetil fosfato}$
R00248	GLUDy	$\text{L-Glutamato} + NADP^+ + H_2O \Leftrightarrow$ $2\text{-Oxoglutarato} + \text{Amoniac} + NADPH + H^+$
R00253	GLNS	$ATP + \text{L-Glutamato} + \text{Amoniac} \Leftrightarrow$ $ADP + \text{Ortofosfato} + \text{L-Glutamina}$
R00256	GLUN	$\text{L-Glutamina} + H_2O \Leftrightarrow \text{L-Glutamato} + \text{Amoniac}$

ID KEGG	ID BiGG	Ecuación
R00267	ICDH _{yr}	Isocitrato + $NADP^+$ \Leftrightarrow 2-Oxoglutarato + CO_2 + $NADPH$ + H^+
R00299	-	ATP + D-Glucosa \Leftrightarrow ADP + D-Glucosa 6-fosfato
R00315	ACK _r	ATP + Acetato \Leftrightarrow ADP + Acetil fosfato
R00341	PPCK	ATP + Oxaloacetato \Leftrightarrow ADP + Fosfoenolpiruvato + CO_2
R00342	MDH	(S)-Malato + NAD^+ \Leftrightarrow Oxalacetato + $NADH$ + H^+
R00345	PPC	Ortofosfato + Oxalacetato \Leftrightarrow H_2O + Fosfoenolpiruvato + CO_2
R00351	CS	Citrato + CoA \Leftrightarrow Acetil-CoA + H_2O + Oxalacetato
R00405	SUCOAS	ATP + Succinato + CoA \Leftrightarrow ADP + Ortofosfato + Succinil-CoA
R00472	MALS	(S)-Malato + CoA \Leftrightarrow Acetil-CoA + H_2O + Glioxilato
R00479	ICL	Isocitrato \Leftrightarrow Succinato + Glioxilato
R00658	ENO	2-fosfo-D-glicerato \Leftrightarrow Fosfoenolpiruvato + H_2O
R00704	LDH_D	(R)-Lactato + NAD^+ \Leftrightarrow Piruvato + $NADH$ + H^+
R00754	ALCD2x	Etanol + NAD^+ \Leftrightarrow Acetaldehído + $NADH$ + H^+
R00756	PFK	ATP + D-Fructosa 6-fosfato \Leftrightarrow ADP + D-Fructosa 1,6-bifosfato
R00762	FBP	D-Fructosa 1,6-bifosfato + H_2O \Leftrightarrow D-Fructosa 6-fosfato + Ortofosfato
R00771	PGI	D-Glucosa 6-fosfato \Leftrightarrow D-Fructosa 6-fosfato
R00835	G6PDH2r	D-Glucosa 6-fosfato + $NADP^+$ \Leftrightarrow D-Glucono-1,5-lactona 6-fosfato + $NADPH$ + H^+
R01015	TPI	D-Gliceraldehído 3-fosfato \Leftrightarrow Glicerona fosfato
R01056	RPI	D-Ribosa 5-fosfato \Leftrightarrow D-Ribulosa 5-fosfato
R01061	GAPD	D-Gliceraldehído 3-fosfato + Ortofosfato + NAD^+ \Leftrightarrow 3-fosfo-D-glicerol fosfato + $NADH$ + H^+

ID KEGG	ID BiGG	Ecuación
R01067	TKT2	D-Fructosa 6-fosfato + D-Gliceraldehído 3-fosfato \Leftrightarrow D-Eritrosa 4-fosfato + D-Xilulosa 5-fosfato
R01068	FBA	D-Fructosa 1,6-bifosfato \Leftrightarrow Glicerona fosfato + D-Gliceraldehído 3-fosfato
R01082	FUM	(S)-Malato \Leftrightarrow Fumarato + H_2O
R01325	ACONTa	Citrato \Leftrightarrow cis-Aconitato + H_2O
R01512	PGK	$ATP + 3\text{-Fosfo-D-glicerato} \Leftrightarrow$ $ADP + 3\text{-Fosfo-D-glicerol fosfato}$
R01518	PGM	2-Fosfo-D-glicerato \Leftrightarrow 3-Fosfo-D-glicerato
R01528	GND	6-Fosfo-D-gluconato + $NADP^+ \Leftrightarrow$ D-Ribulosa 5-fosfato + $CO_2 + NADPH + H^+$
R01529	RPE	D-Ribulosa 5-fosfato \Leftrightarrow D-Xilulosa 5-fosfato
R01641	TKT1	Sedoheptulosa 7-fosfato + D-Gliceraldehído 3-fosfato \Leftrightarrow D-Ribosa 5-fosfato + D-Xilulosa 5-fosfato
R01900	ACONTb	Isocitrato \Leftrightarrow cis-Aconitato + H_2O
R02035	PGL	D-Glucono-1,5-lactona 6-fosfato + H_2O \Leftrightarrow 6-Phospho-D-gluconato
R02164	SUCDi	Quinona + Succinato \Leftrightarrow Hydroquinona + Fumarato
R08549	AKGDH	2-Oxoglutarato + $CoA + NAD^+ \Leftrightarrow$ Succinil-CoA + $CO_2 + NADH + H^+$
R08575	TALA	Sedoheptulosa 7-fosfato + D-Gliceraldehído 3-fosfato \Leftrightarrow D-Eritrosa 4-fosfato + D-Fructosa 6-fosfato

Tabla 8.6: Reacciones presentes en red metabólica del metabolismo central

8.6. Casos de estudio adicionales

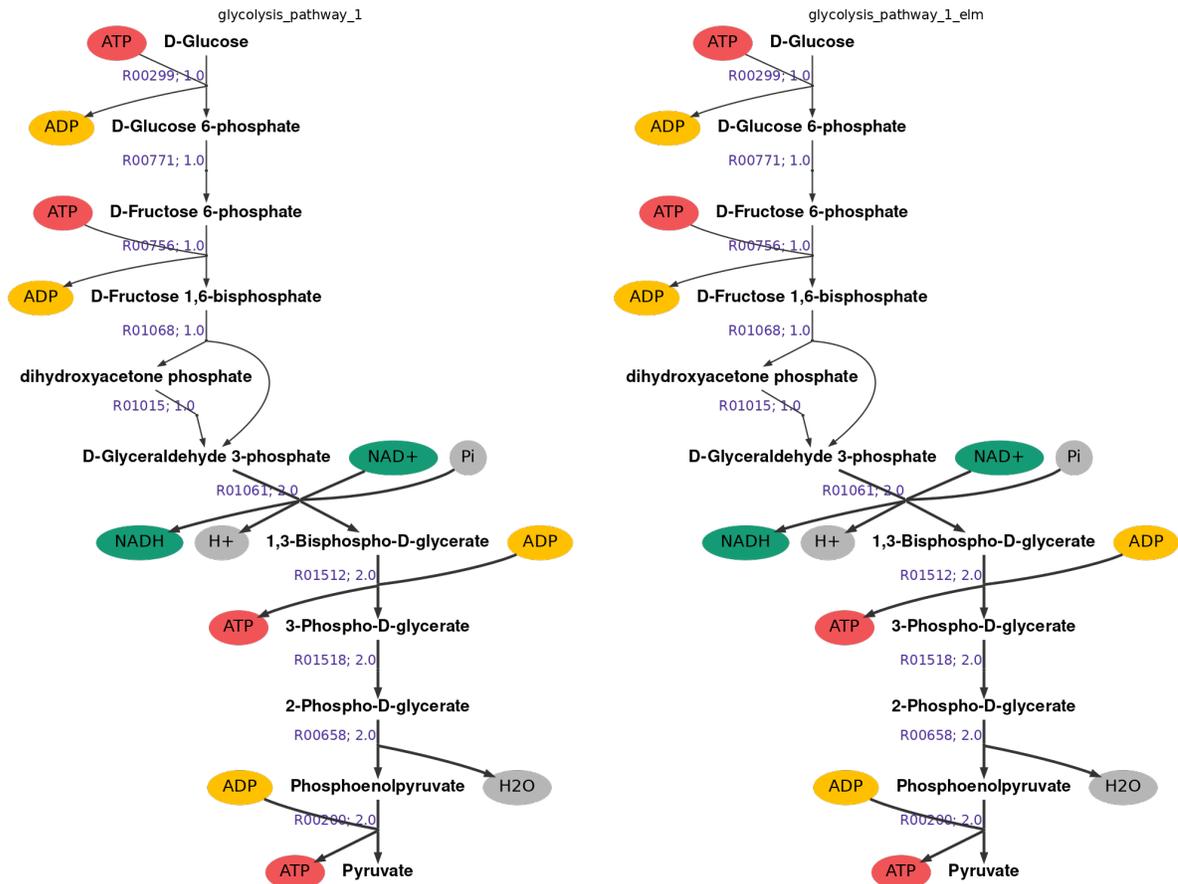
8.6.1. Producción de piruvato a partir de glucosa

Para este caso se espera obtener como resultado la ruta de la glicólisis, por lo que los límites para entradas y salidas del algoritmo están definidos como aparece en la Tabla 8.7

Tabla 8.7: Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de piruvato a partir de glucosa

Compuesto	$LB(\frac{mmol}{gr S_0 \cdot hr})$	$UB(\frac{mmol}{gr S_0 \cdot hr})$
Glucosa	-1	-1
Piruvato	2	2
<i>ATP</i>	2	2
<i>ADP</i>	-2	-2
<i>NAD⁺</i>	-2	-2
<i>NADH</i>	2	2
<i>H⁺</i>	2	2
<i>H₂O</i>	2	2
<i>Pi</i>	-2	-2

En la Figura 8.4 se pueden apreciar los resultados del algoritmo y en la Tabla 8.8 se observan las cargas enzimáticas obtenidas.



(a) Ruta obtenida minimizando flujos (flujo total: $15 \frac{mmol}{gr S_0 \cdot hr}$).

(b) Ruta obtenida minimizando carga enzimática (flujo total: $15 \frac{mmol}{gr S_0 \cdot hr}$).

Figura 8.4: Rutas para la producción de piruvato a partir de glucosa, vía glicólisis

Tabla 8.8: Cargas enzimáticas para la producción de piruvato a partir de glucosa

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr S_0}$)
Ruta 1	R00200	$-8.35 \cdot 10^{-6}$
	R00299	11.3
	R00658	1.47
	R00756	1.80
	R00771	$1.48 \cdot 10^{-1}$
	R01015	-7.19
	R01061	3.15
	R01068	2.19
	R01512	$-1.40 \cdot 10^{-4}$
	R01518	-9.44
	Carga total	3.43

Al minimizar tanto flujos como carga enzimática se obtiene la misma ruta. Este resultado era esperable ya que la glicólisis es la ruta más directa entre ambos compuestos dada la estequiometría general de la Tabla 8.7.

Como se observa en la Tabla 8.8, hay cargas enzimáticas netas que son negativas. Dos de ellas corresponden a reacciones que involucran *ATP* y su carga enzimática es cercana a cero siendo, a lo menos, tres órdenes de magnitud menor que las demás reportadas. Estos valores de carga neta no impiden que haya flujo por dichas reacciones. Las otras dos reacciones con carga enzimática neta negativa corresponden a reacciones reversibles, aunque su magnitud sea mayor tampoco impide el paso del flujo por dichas reacciones al momento de plantear las restricciones en el algoritmo.

Podría generarse un desvío por la ruta de las pentosas fosfato, volviendo a unirse en la producción de gliceraldehído 3-fosfato, pero al no entregarle los compuestos a la entrada o salida necesarios para utilizar esta ruta (CO_2 , $NADP^+/NADPH$, entre otros), esta posible solución no es generada.

8.6.2. Producción de piruvato a partir de distintas fuentes (límites holgados)

Para este caso de estudio se consideraron los límites de entrada y salida especificados en la Tabla 8.9, los cuales responden a la estequiometría de las tres posibles rutas a obtener y la producción entre uno y dos moles de piruvato, dependiendo del sustrato inicial.

Tabla 8.9: Rangos de tasas de producción o consumo para los compuestos de entrada y salida, producción de piruvato a partir de distintas fuentes de carbono

Compuesto	$LB(\frac{mmol}{gr S_0 \cdot hr})$	$UB(\frac{mmol}{gr S_0 \cdot hr})$
Glucosa	-1	0
Malato	-1	0
Lactato	-1	0
Piruvato	1	2
NAD^+	-2	0
$NADH$	0	2
ATP	0	2
ADP	-2	0
$NADP^+$	-2	0
$NAPDH$	0	2
H^+	1	2
H_2O	0	1
CO_2	0	1
P_i	0	2

Los resultados obtenidos por el algoritmo minimizando flujos se pueden apreciar en la Figura 8.5.

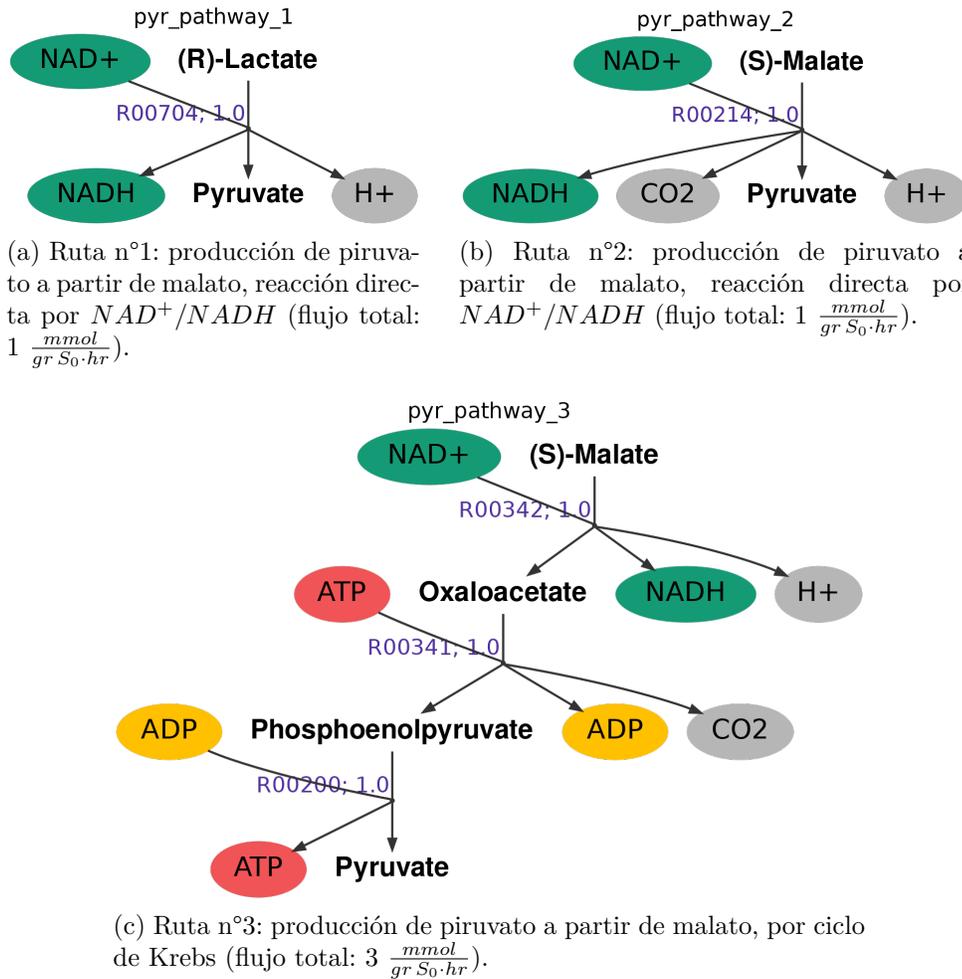


Figura 8.5: Resultados del algoritmo en prueba de producción de 1-2 moles de piruvato minimizando flujos

Las rutas obtenidas por el algoritmo solo utilizan malato o lactato como fuentes de carbono. Al ser rutas más cortas es posible que se de por terminada la optimización antes de llegar a utilizar glucosa, puesto que conlleva un mayor flujo por su mayor cantidad de compuestos involucrados y por su mayor rendimiento.

Las dos primeras rutas corresponden a las reacciones directas esperadas de sus respectivos sustratos iniciales, mientras que la tercera también utiliza malato pero pasa por compuestos intermedios que conectan el ciclo de Krebs con la glicólisis. En las tres rutas se utiliza solo $NAD^+/NADH$, no se obtienen las reacciones equivalentes con $NADP^+/NADPH$ aunque conllevan el mismo flujo que las soluciones entregadas.

Siguiendo la misma lógica, se cambia el objetivo a minimizar la carga enzimática. Se mantienen los límites de la Tabla 8.9. En la Figura 8.6 aparecen las rutas obtenidas con sus respectivas cargas enzimáticas netas en la Tabla 8.10.

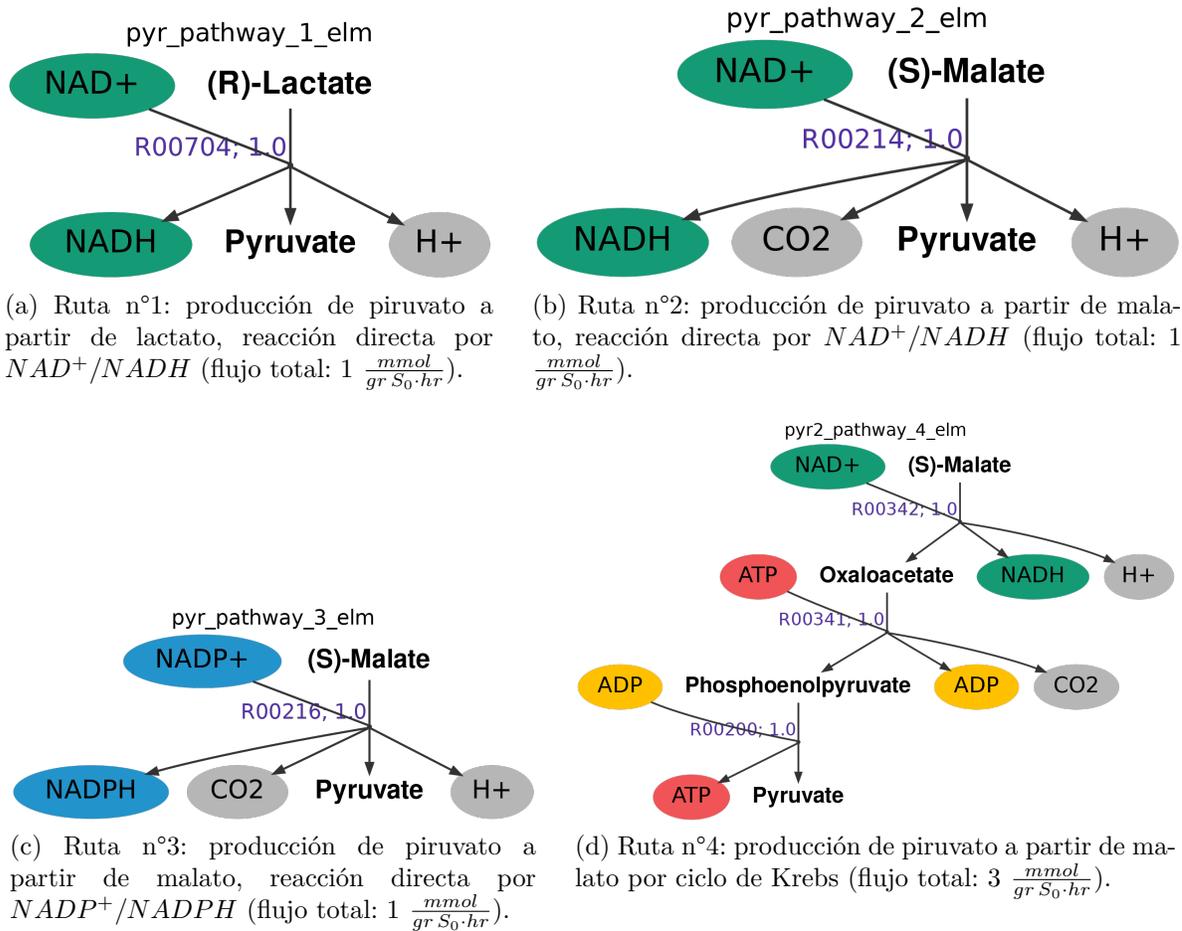


Figura 8.6: Resultados del algoritmo en prueba de producción de 1-2 moles de piruvato minimizando carga enzimática

Tabla 8.10: Cargas enzimáticas para la producción de piruvato a partir de distintas fuentes, límites holgados. (***) : el optimizador aproxima este valor a cero debido a que su orden de magnitud es mucho menor en comparación a los demás.

Solución	ID reacción	Carga enzimática ($\frac{mg \text{ enzima}}{gr S_0}$)
Ruta 1	R00704	4.37
Ruta 2	R00214	5.04
Ruta 3	R00216	12.6
Ruta 4	R00200	***
	R00341	$8.11 \cdot 10^{-1}$
	R00342	$2.59 \cdot 10^{-1}$
	Carga total	1.07

Las primeras dos rutas coinciden con lo visto al minimizar flujos. Esta vez se obtiene como solución la ruta 3 que utiliza malato como fuente de carbono y $NADP^+$ como transportador de electrones, solución que no se obtuvo al minimizar flujos. Por otro lado, la cuarta ruta, que es la menos óptima de las cuatro, hay un mayor uso de cofactores involucrados (ATP y

ADP). Se destaca de esta última ruta que la reacción R00200 presenta una carga enzimática nula, pero esta reacción se vio que su carga enzimática neta tomaba un valor con órdenes de magnitud muy pequeños en el primer caso de estudio para el metabolismo central. Es probable que se considerara despreciable para el optimizador y se aproximara a cero.

Aquí la suma de las cargas netas de la ruta más óptima no coincide con ser la más baja de todas las rutas, pero lo que se minimiza es la carga enzimática de cada reacción en ambos sentidos (directo e inverso). Por ello, al hacer el balance es posible que la carga neta total de una ruta menos óptima sea de menor valor que otra más óptima.

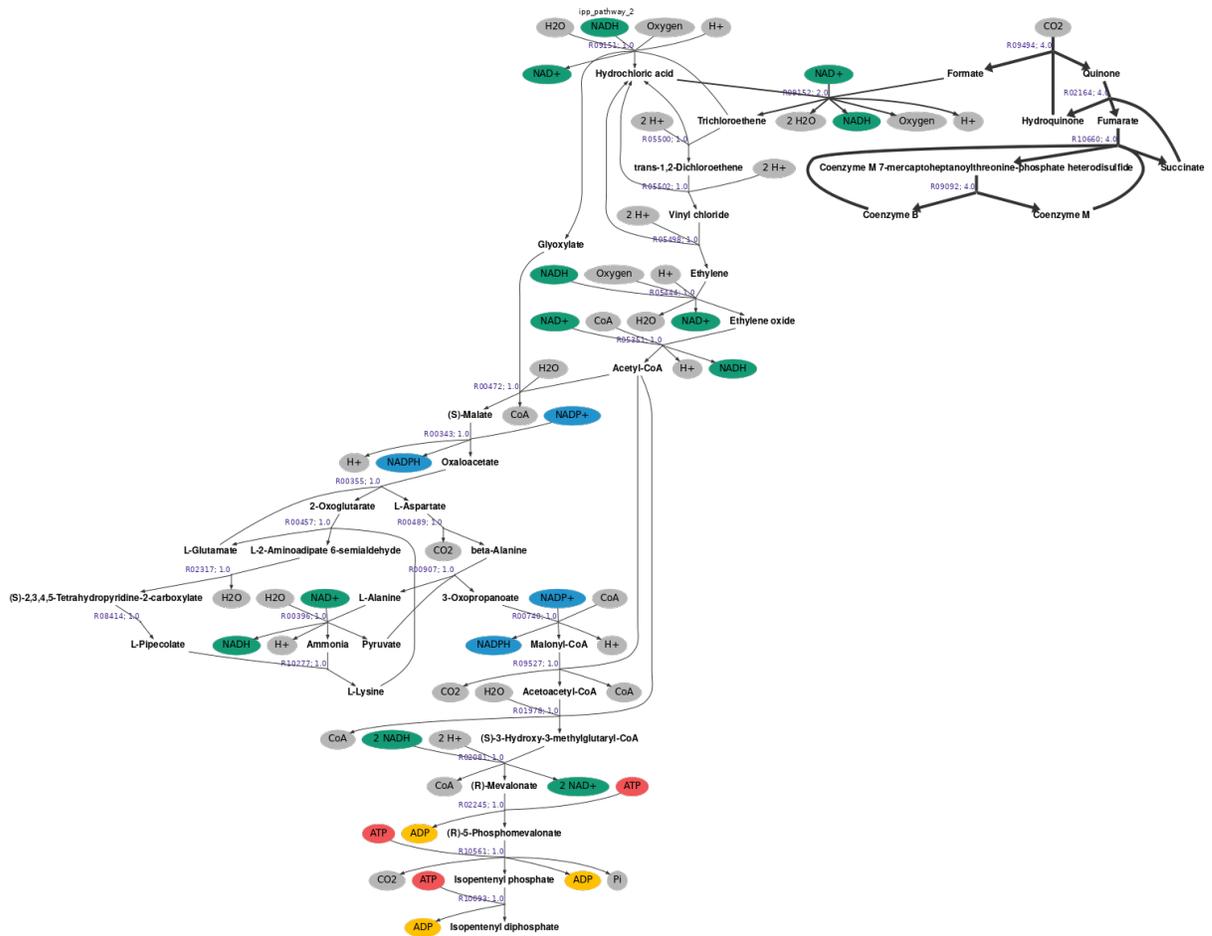


Figura 8.8: Producción de isopentenil difosfato a partir de acetil-CoA minimizando flujos (Ruta 2)

8.6.4. Rutas adicionales para la producción de *IPP* minimizando flujos

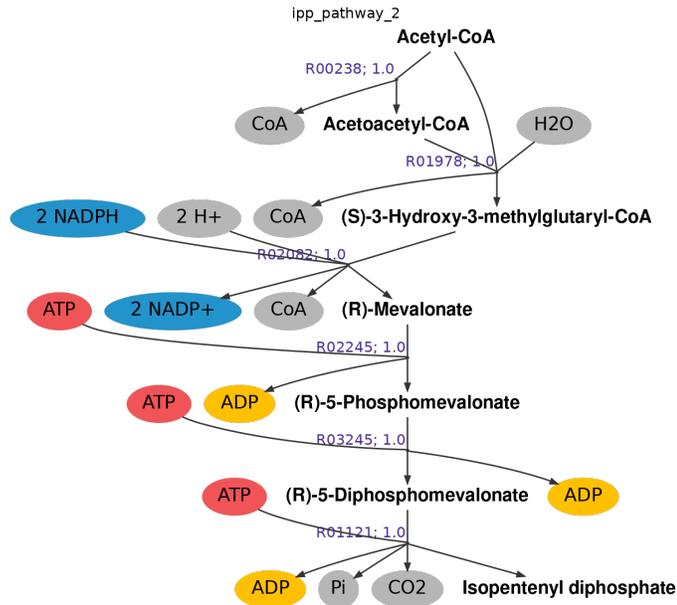


Figura 8.9: Ruta 2: producción de *IPP* a partir de Acetil-coA minimizando flujos (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).

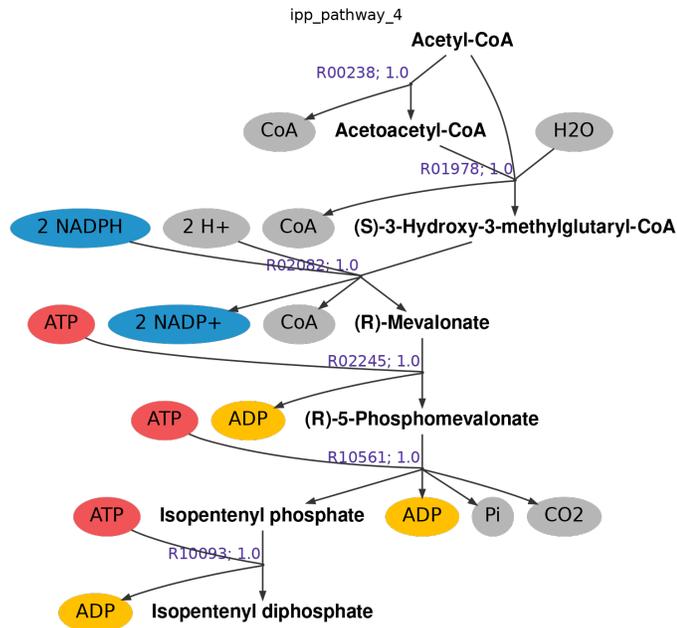


Figura 8.10: Ruta 4: producción de *IPP* a partir de Acetil-coA minimizando flujos (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).

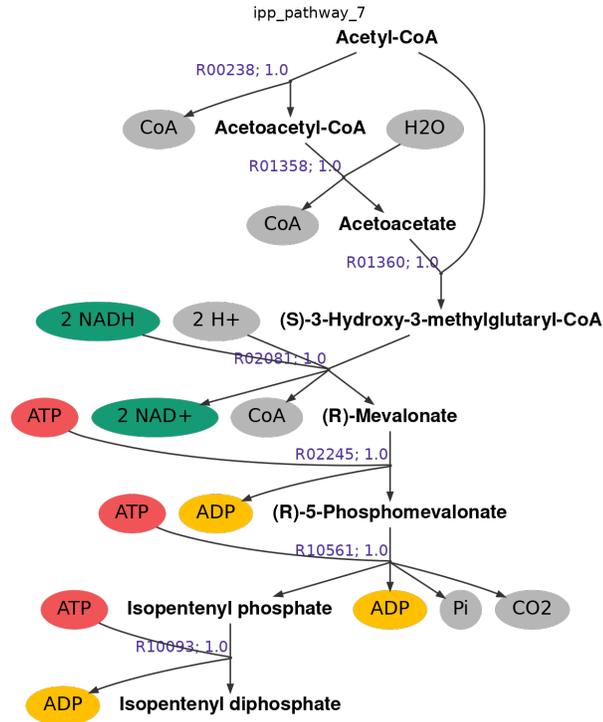


Figura 8.11: Ruta 7: producción de *IPP* a partir de Acetil-coA minimizando flujos (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).

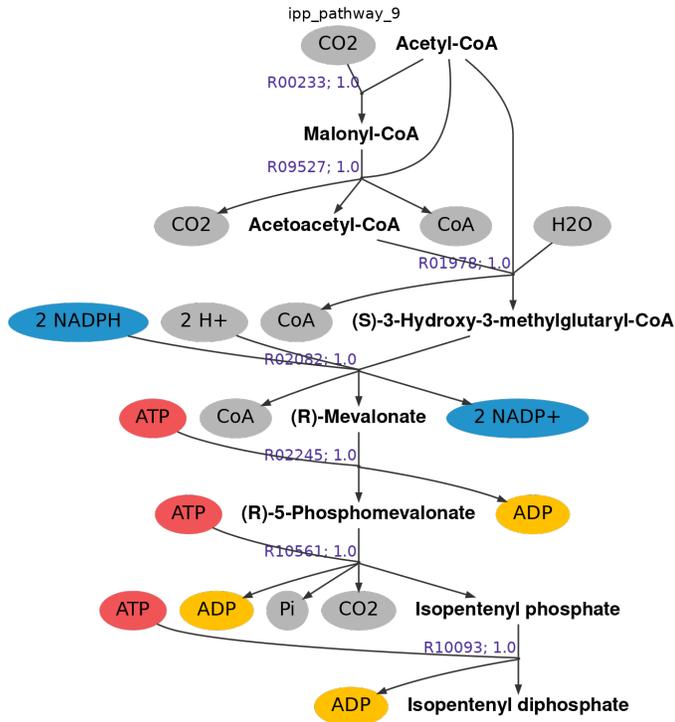


Figura 8.12: Ruta 9: producción de *IPP* a partir de Acetil-coA minimizando flujos (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).

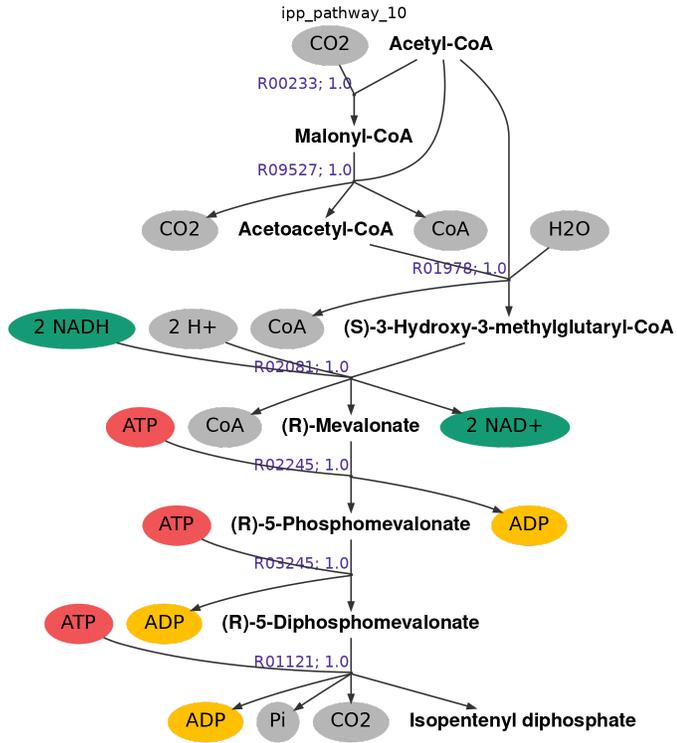


Figura 8.13: Ruta 10: producción de *IPP* a partir de Acetil-coA minimizando flujos (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).

8.6.5. Producción de mevalonato a partir de acetil-CoA

En la Tabla 8.11 se encuentran los límites de la estequiometría impuesta al algoritmo para este caso de estudio. Se mantuvo la estequiometría de esta sección de la ruta, dando más libertad en el uso de agua, iones, *ATP* y *ADP*.

Tabla 8.11: Rangos de tasas de producción o consumo para los compuestos de entrada y salida para la producción de Mevalonato a partir de Acetil-CoA

Compuesto	$LB(\frac{mmol}{gr\ S_0 \cdot hr})$	$UB(\frac{mmol}{gr\ S_0 \cdot hr})$
Acetil-CoA	-3	-3
Mevalonato	1	1
<i>NAD</i> ⁺	0	2
<i>NADH</i>	-2	0
<i>NADP</i> ⁺	0	2
<i>NADPH</i>	-2	0
<i>ADP</i>	-1	1
<i>ATP</i>	-1	1
<i>H</i> ₂ <i>O</i>	-1	1
<i>CoA</i>	3	3
<i>H</i> ⁺	-2	2

8.6.5.1. Producción de mevalonato minimizando flujos

Previamente se intentó forzar el uso de *ATP* en las rutas a obtener de forma similar a lo reportado en la literatura [16], pero no se obtuvo ninguna solución factible. Los resultados del algoritmo al minimizar flujos no variaron en relación a los obtenidos en el caso de producción de *IPP*. Las rutas obtenidas en esta prueba se pueden observar desde las figura 8.14 hasta la figura 8.23.

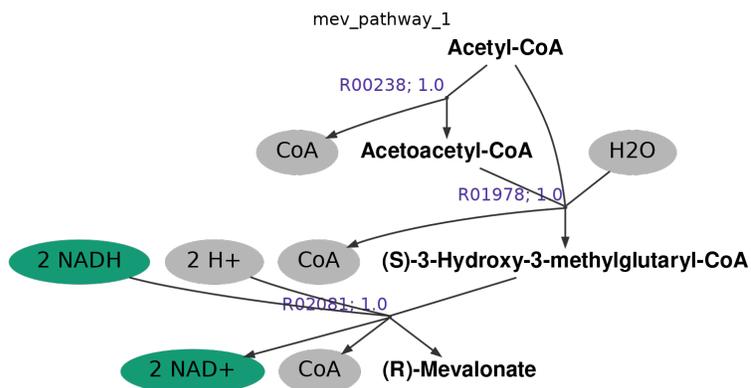


Figura 8.14: Ruta 1: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $3 \frac{mmol}{gr\ S_0 \cdot hr}$).

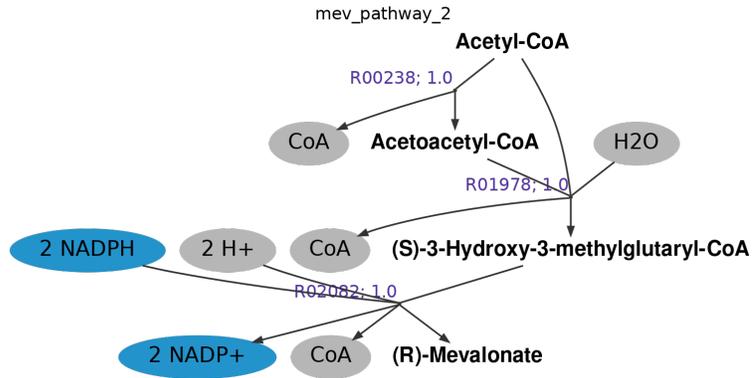


Figura 8.15: Ruta 2: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $3 \frac{mmol}{gr S_0 \cdot hr}$).

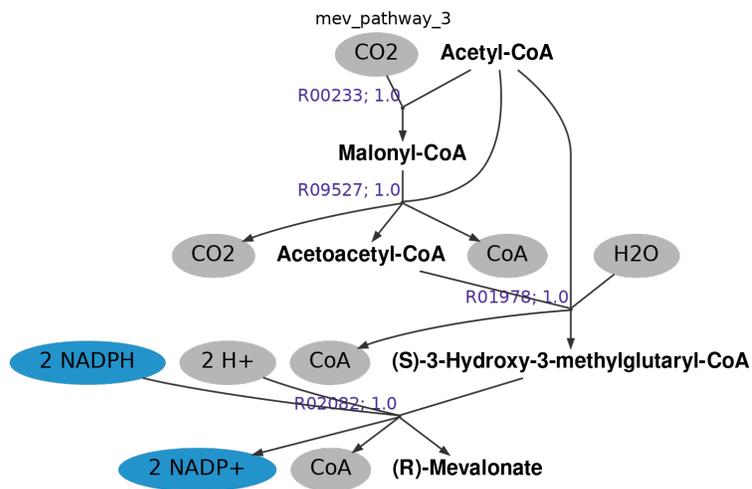


Figura 8.16: Ruta 3: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).

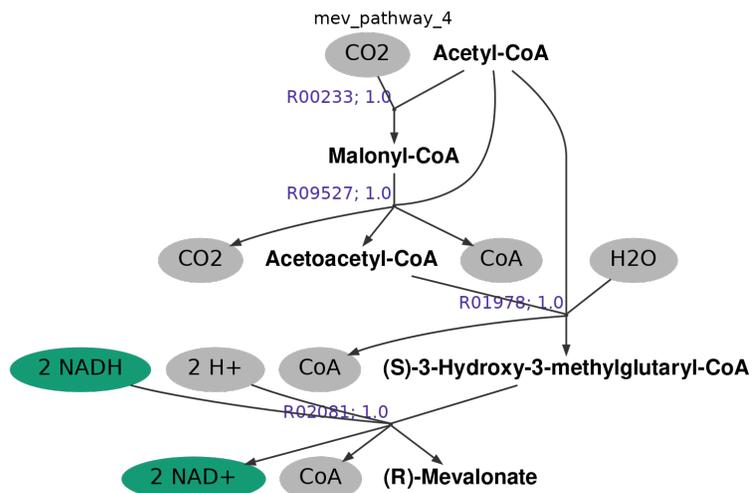


Figura 8.17: Ruta 4: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).

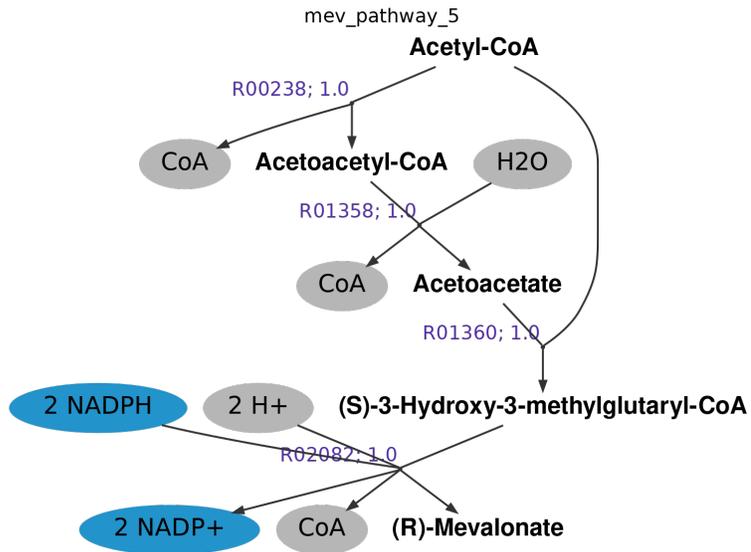


Figura 8.18: Ruta 5: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).

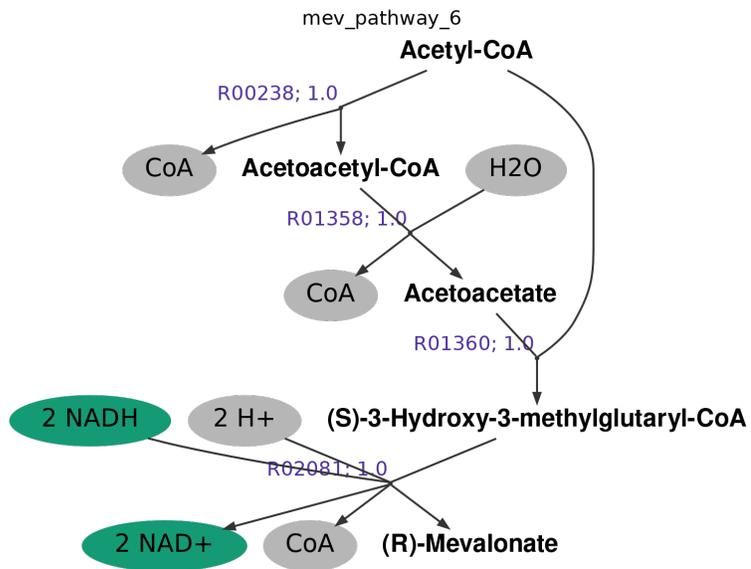


Figura 8.19: Ruta 6: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $4 \frac{mmol}{gr S_0 \cdot hr}$).

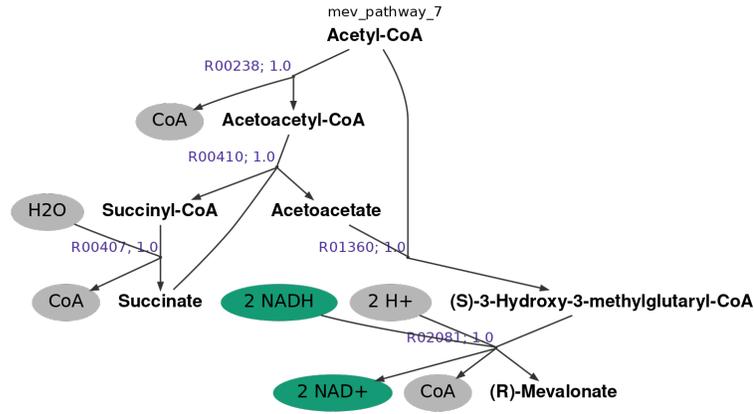


Figura 8.20: Ruta 7: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).

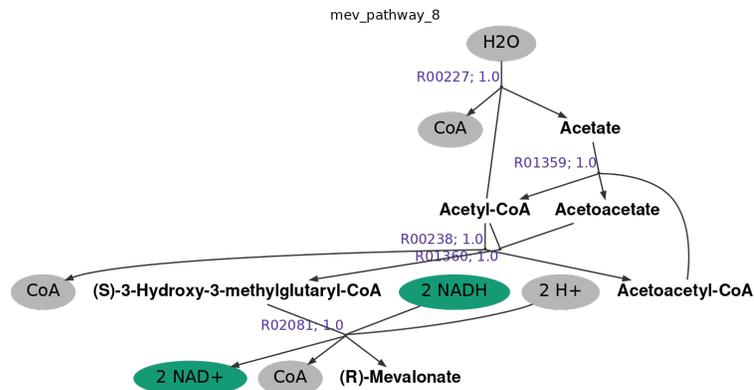


Figura 8.21: Ruta 8: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).

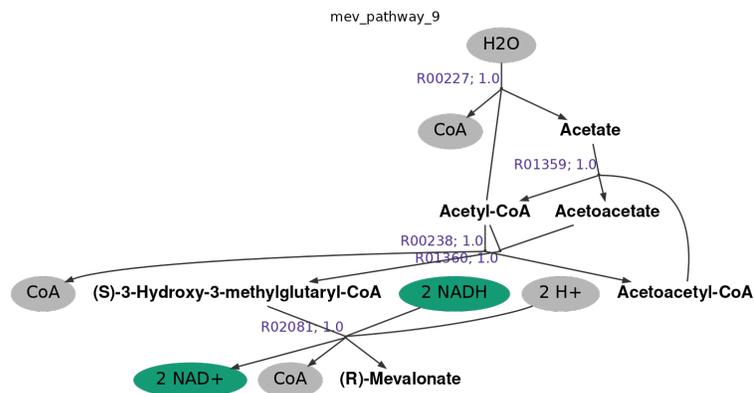


Figura 8.22: Ruta 9: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).

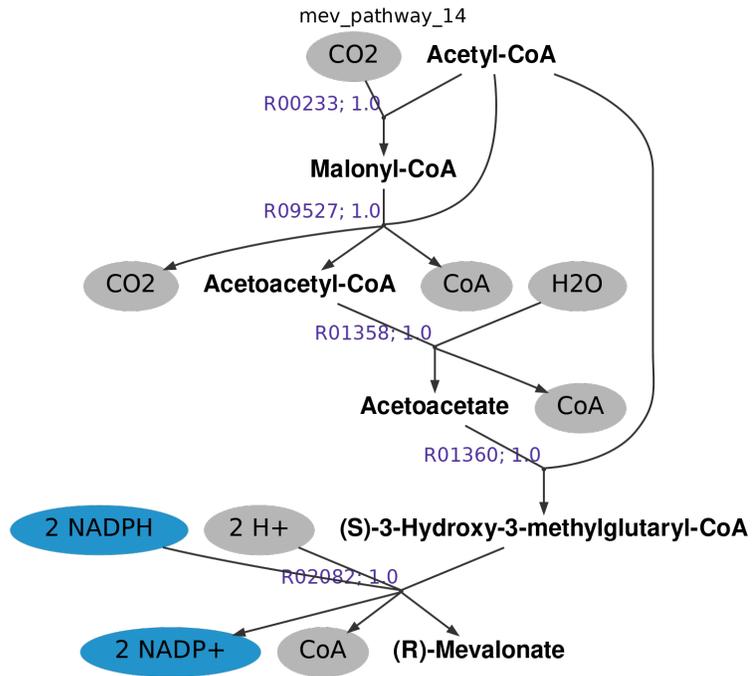


Figura 8.23: Ruta 10: producción de Mevalonato a partir de Acetil-coA minimizando flujos (flujo total: $5 \frac{mmol}{gr S_0 \cdot hr}$).

8.6.5.2. Producción de mevalonato minimizando carga enzimática

Desde la Figura 8.24 hasta la 8.28 se pueden apreciar las rutas obtenidas, mientras que en las Tablas 8.12 a la 8.16, aparecen sus respectivas cargas enzimáticas. Estos resultados fueron generados a partir de la misma estequiometría descrita en la Tabla 8.11

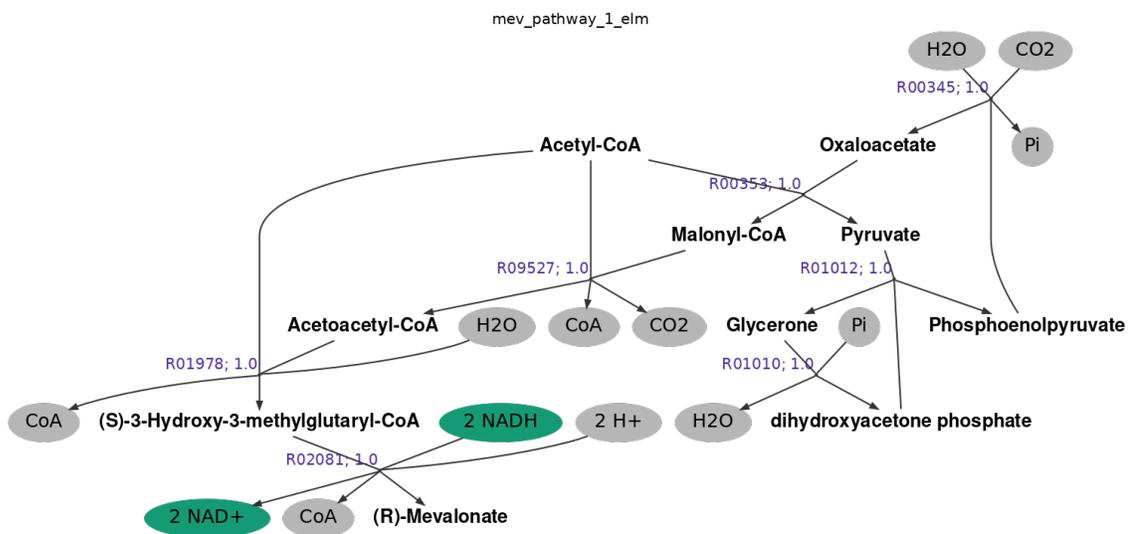


Figura 8.24: Ruta 1: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 8.12: Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 1)

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr S_0}$)
Ruta 1	R00345	$-1.90 \cdot 10^{-4}$
	R00353	-2.57
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.96
	Carga total	-3.54

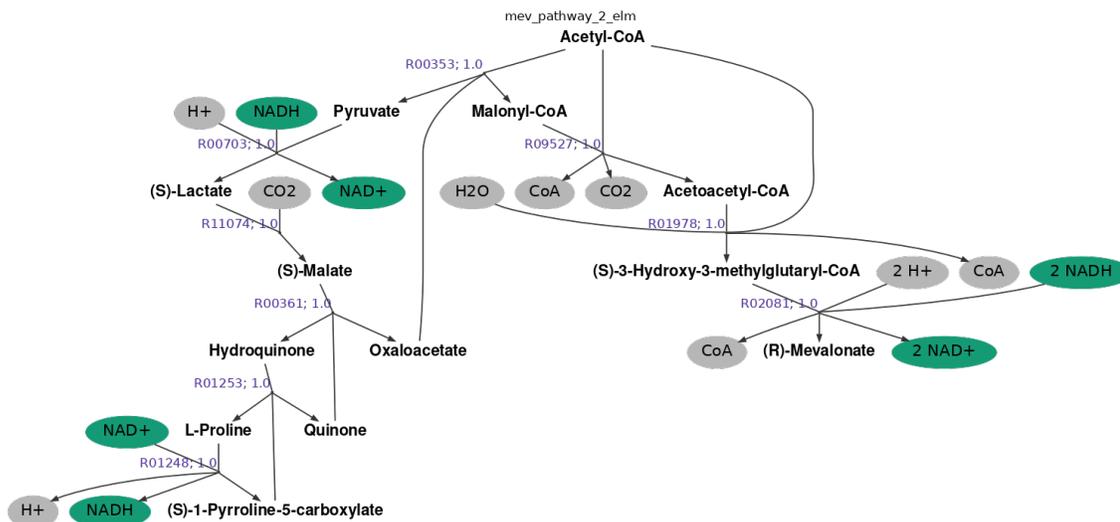


Figura 8.25: Ruta 2: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $9 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 8.13: Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 2)

Solución	ID reacción	Carga enzimática ($\frac{mg \text{ enzima}}{gr S_0}$)
Ruta 2	R00353	-2.57
	R00703	$-6.32 \cdot 10^{-5}$
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.96
	R11074	-0.14
	Carga total	-3.68

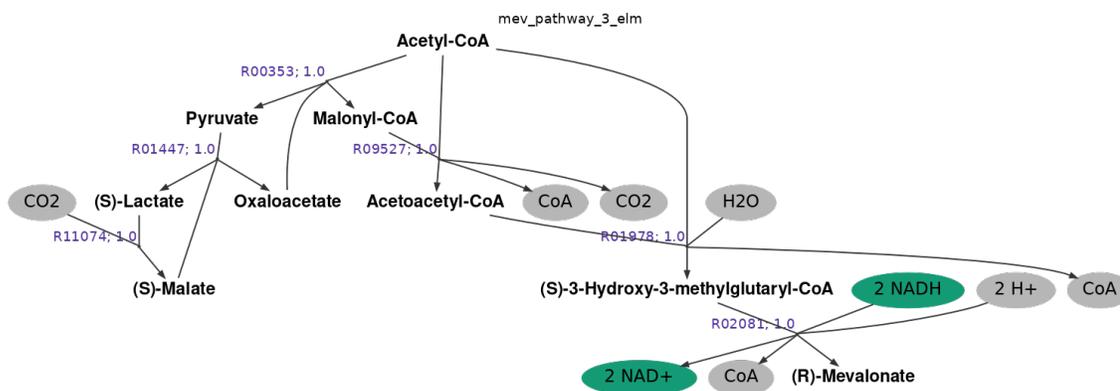


Figura 8.26: Ruta 3: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $6 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 8.14: Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 3)

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr S_0}$)
Ruta 3	R00353	-2.57
	R01447	-0.23
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.96
	R11074	-0.14
	Carga total	-3.91

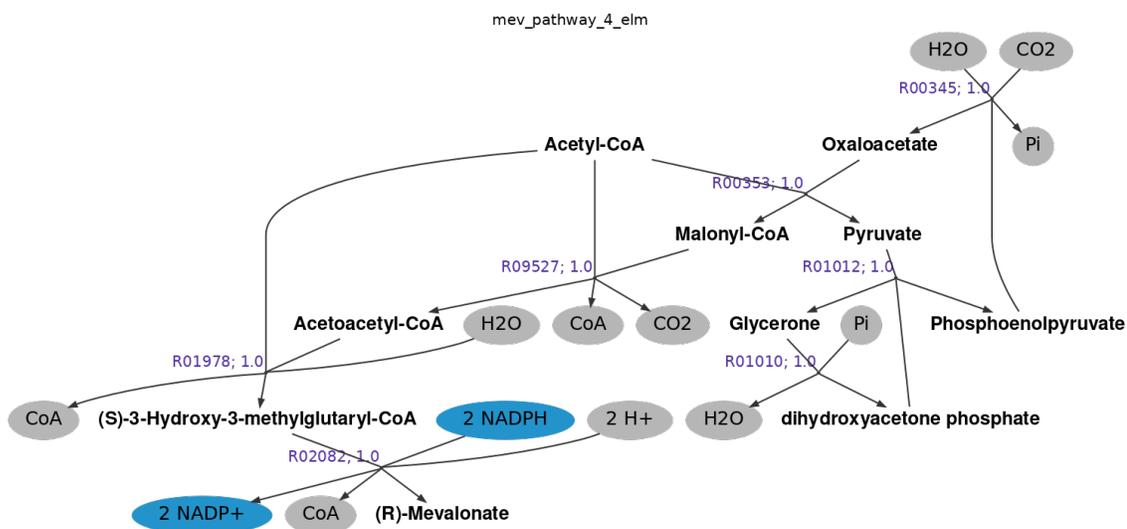


Figura 8.27: Ruta 4: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $7 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 8.15: Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 4)

Solución	ID reacción	Carga enzimática ($\frac{mg\ enzima}{gr S_0}$)
Ruta 4	R00345	$-1.90 \cdot 10^{-4}$
	R00353	-2.57
	R01978	$-6.48 \cdot 10^{-3}$
	R02082	-1.57
	Carga total	-4.34

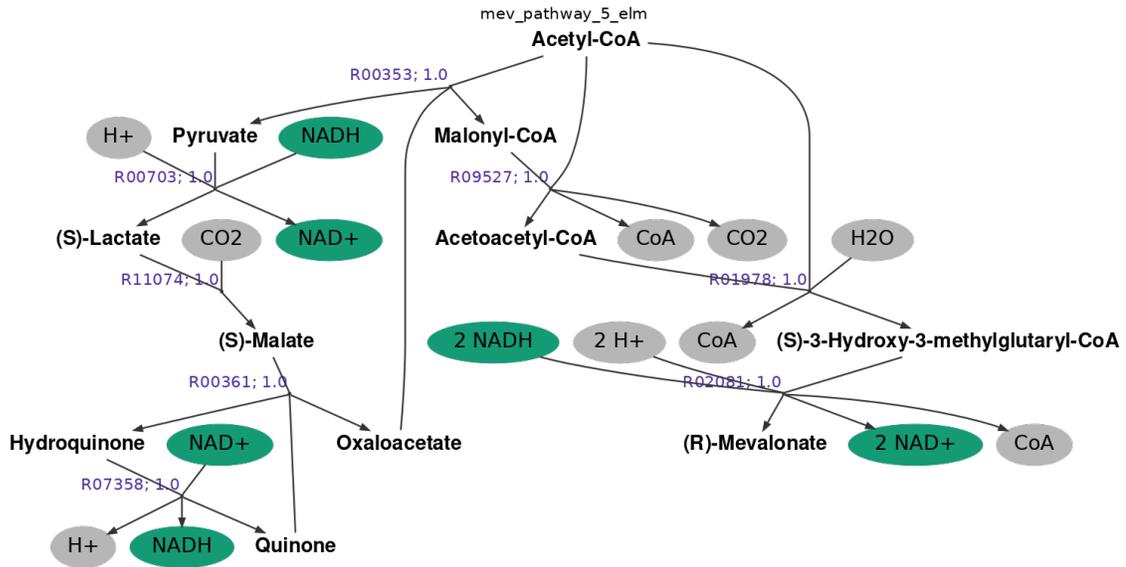


Figura 8.28: Ruta 5: producción de Mevalonato a partir de Acetil-coA minimizando carga enzimática (flujo total: $8 \frac{mmol}{gr S_0 \cdot hr}$).

Tabla 8.16: Cargas enzimáticas para la producción de Mevalonato a partir de acetil CoA (Ruta 5)

Solución	ID reacción	Carga enzimática ($\frac{mg \text{ enzima}}{gr S_0}$)
Ruta 5	R00353	-2.57
	R00703	$-6.32 \cdot 10^{-5}$
	R01978	$-6.48 \cdot 10^{-3}$
	R02081	-0.96
	R11074	-0.14
	Carga total	-3.68

Todas las rutas obtenidas forman acetoacetyl-CoA a partir de malonil-CoA y acetil-CoA (R09527), el cual es formado por distintas combinaciones de reacciones en cada solución. Nuevamente se observa la limitante de la falta de actividades específicas, pues las variantes entre cada ruta obtenida corresponden en su mayoría a reacciones con carga enzimática nula por falta de este parámetro. Las reacciones para las que si se cuenta con una carga enzimática no nula tienen sentido negativo y, en algunos casos, de órdenes de magnitud menores. Este comportamiento ya se había visto para el mismo conjunto de reacciones en el caso de producción de *IPP*

Las rutas 1 y 4 difieren solo en la reacción que requiere transportador de electrones, *NADH/NAD⁺* y *NADPH/NADP⁺*, respectivamente. Para la formación de malonil-CoA, las rutas 2 y 5 utilizan transportadores de electrones pero manteniendo la estequiometría impuesta, consumiendo en una reacción lo producido en otra. Las reacciones que tienen órdenes de magnitud más pequeños son las mismas ya mencionadas en los resultados para la producción de *IPP*.

Si bien entre cada ruta obtenida se presentan pequeñas variaciones, no se obtuvo soluciones distintas a las ya revisadas en el caso de estudio para producir *IPP* a partir de acetyl-CoA. Se mantiene el mismo problema de considerar reacciones sin actividad específica para la minimización de carga enzimática, pero al minimizar flujos tampoco se aprecian variaciones en comparación a las soluciones para la producción de *IPP* con el mismo objetivo.