

## ANEXO I

```
%Función cuyas entradas son una tabla con los nombres de las reacciones de
%interés, y el modelo. Selecciona algunas reacciones de la matriz resultante
%del IOMA a partir de esta matriz del IOMA, el modelo y la tabla con los
%nombres de las reacciones de interés.
function [matrix_reac,imp_reac_idx,Mat_results] = Selectreac (array_strings,model,IOMA_final_matrix)

%Calcula la longitud de la tabla de reacciones
m = length (array_strings)

%Crea variables vacías
matrix_reac = [];
mat_nf = [];

%Se comienza un bucle que se repite tantas veces como n° de reacciones
% en la tabla.
for i = 1:m
    %Busca el nombre de cada reacción en el modelo (en el atributo
    %reacciones), y almacena el índice (la posición en la que se encuentra)
    %en la variable idx.
    idx = find(strcmp(array_strings(i), model.reactions));
    %Guarda el nombre de las reacciones en una variable.
    nombre = model.reactions(idx);
    % Almacena en la matriz, todos los datos anteriores, y para este bucle
    % el nombre, y el índice.
    matrix_reac = [matrix_reac; nombre,idx];
    %Siguiendo la misma estructura que la línea anterior pero guardando en
    %la misma línea y en distintas columnas, el nombre de la tabla, el del
    %modelo y el índice se comprueba que el programa está cogiendo las
    %reacciones correctas.
end

% Una vez terminado el bucle, el vector de índices corresponde a la segunda
% matriz de la matriz de reacciones de interés. En este comando se
% selecciona esta segunda columna y se almacena en un vector de nombre
% imp_reac_idx.
imp_reac_idx = matrix_reac(:,2);
P = cell2mat(imp_reac_idx)
n = length (P);
Mat_results = [];
for i = 1:n
    Mat_results = [Mat_results; IOMA_final_matrix(P(i,:)]);
end
end
```

Figura 1. Función en Matlab para escoger de la matriz resultante de IOMA, los valores de aquellas reacciones que son de interés.

```

1  %El programa pide al usuario el numero de reacciones de intercambio y el
2  %numero de condiciones de la simulación.
3  a = input('How many exchange reactions are in the model ? ');
4  b = input('How many conditions are you going to use ? ');
5  %Se crea una matriz vacia y una matriz de ceros
6  ex_flux = zeros(a,b);
7  ex_rxns = [];
8
9  %Bucle que se repite tantas veces como número de reacciones
10 for i = 1:a
11     %Pide al usuario el indice de las reacciones de intercambio.
12     reac_idx = input('Which is the index of the reaction? ');
13     %Almacena estos indices en un vector fila.
14     ex_rxns = [ex_rxns,reac_idx];
15     %Bucle que se repite tantas veces como número de condiciones
16     for j = 1:b
17         %Pide el flujo para cada reacción y condición al usuario
18         flux = input(['Give me the flux of the exchange reaction ' int2str(i) ' in condition ' int2str(j) ' : ']);
19         %Almacena estos flujos en una matriz.
20         ex_flux(i,j)= flux;
21     end
22 end
23
24 %Imprime en pantalla el vector de indices.
25 disp ' This is your vector of indexes of exchange reactions : '
26 ex_rxns
27
28 %Comando que pregunta al usuario si este vector es correcto y si no lo es
29 %entra en un bucle que pregunta en que columna se encuentra el error y se
30 %pide el nuevo valor. Este bucle no finaliza hasta que el usuario no indica
31 %que el vector es correcto.
32 c = input('Is everything okey? 0(yes) or 1(no) ');
33 while c==1
34     n = input('In which column is your wrong data? ');
35     ex_rxns(1,n) = input ('Write the correct value : ');
36     disp ' This is your vector of indexes of exchange reactions : '
37     ex_rxns
38
39     c = input('Is everything okey? 0(yes) or 1(no)');
40 end
41
42 %Imprime en pantalla la matriz de flujos de intercambio.
43 disp ' This is your matrix if exchange fluxes : '
44 ex_flux
45
46 %Comando que pregunta al usuario si esta matriz es correcta y si no lo es
47 %entra en un bucle que pregunta en que fila y columna se encuentra el error
48 %y se pide el nuevo valor. Este bucle no finaliza hasta que el usuario no
49 %indique que la matriz es correcta.
50 d = input('Is everything okey? 0(yes) or 1(no) ');
51
52 while d==1
53     m = input('In which row is your wrong data? ');
54     n = input('In which column is your wrong data? ');
55     ex_flux(m,n) = input ('Write the correct value : ');
56     disp ' This is your matrix of exchange fluxes : '
57     ex_flux
58
59     d = input('Is everything okey? 0(yes) or 1(no)');
60 end

```

Figura 2. Programa que ordena los datos de flujo en las reacciones de intercambio a partir de datos tecleados manualmente por el usuario.

```

1      %Comando que pide al usuario el n° de reacciones y lo almacena en una
2      %variable.
3      A = input ('How many conditions are you going to enter ? ');
4      %Crea una variable vacía
5      GR = [];
6      %Un bucle que se recorre tantas veces como número de condiciones
7      for i = 1:A
8          %Pide al usuario el valor de crecimiento para condición.
9          Condition = input(['Give me the growth rate in condition ' int2str(i) ' : ']);
10         %Almacena los datos de crecimiento en un vector fila
11         GR = [GR,Condition];
12     end

```

Figura 3. Programa que ordena los datos de crecimiento a partir de información que aporta el usuario de forma interactiva.

```

1      %Función para organizar los datos en matrices que corresponden a algunas
2      %de las entradas del algoritmo IOMA a partir del modelo y de un archivo
3      %excel que se importa a matlab como 'cell'.
4      %El archivo excel contiene 5 columnas: en la primera se encuentran los
5      %nombres de las reacciones, en la segunda los flujos de intercambio de cada
6      %reacción para la condición 1, en la tercera os flujos de intercambio de ca
7      %reacción para la condición 2, en la cuarta y en la quinta los valores de
8      %crecimiento para la condición 1 y 2, respectivamente.
9      function [ex_rxns,ex_flux,GR] = infunctexchange (input_cell,model)
10     %Almacena el número de filas y columnas del archivo excel en las variables
11     %m y n.
12     [m,n] = size (input_cell);
13     %%Crea variables vacías.
14     ex_flux = [];
15     ex_rxns = [];
16     %Convierte cada columna, a partir de la segunda, de la 'célula' de matlab
17     %en una matriz independiente.
18     A = cell2mat(input_cell(:,2));
19     B = cell2mat(input_cell(:,3));
20     C = cell2mat(input_cell(:,4));
21     D = cell2mat(input_cell(:,5));
22     %Almacena los datos de crecimiento para condición en un vector fila.
23     GR = [C(1,1),D(1,1)];
24     %%Crea un bucle para almacenar los datos tantas veces como filas en el exce
25     %de inicio, es decir, tantas veces como reacciones cuyos datos se tienen.
26
27     for i = 1:m
28         %Busca el índice para los nombres de las reacciones que se encuentran
29         %en la primera columna del excel
30         idx = find(strcmp(input_cell(i,1), model.reactions));
31         %Almacena los índices en un vector fila
32         ex_rxns = [ex_rxns, idx];
33         %El valor de flujo de intercambio para esa reacción en la
34         %condición 1 se almacena en una variable.
35         exflux1 = A(i);
36         %Se ordena cada valor en la matriz de flujos de intercambio donde la
37         %columna uno contiene los flujos de intercambio en la primera condición,
38         %y la columna dos los datos de los flujos de intercambio de la segunda condición
39         ex_flux(i,1) = exflux1;
40         %Flujos de intercambio de cada reacción en la segunda condición.
41         exflux2 = B(i);
42         %Se ordenan los datos de flujos en la misma matriz pero esta vez
43         %para la segunda condición.
44         ex_flux(i,2) = exflux2;
45     end
46 end

```

Figura 4. Programa que ordena los datos de flujo en las reacciones de intercambio y de crecimiento a partir de datos ordenados en un excel.

## ANEXO II

### 1. Modelo juguete 1

El modelo juguete 1 es un modelo que consta de 9 reacciones, 4 de ellas reacciones de intercambio, y 10 metabolitos internos. Por tanto su matriz estequiométrica es una matriz 10x9 y sus vectores de límites superiores e inferiores son dos vectores 9x1.

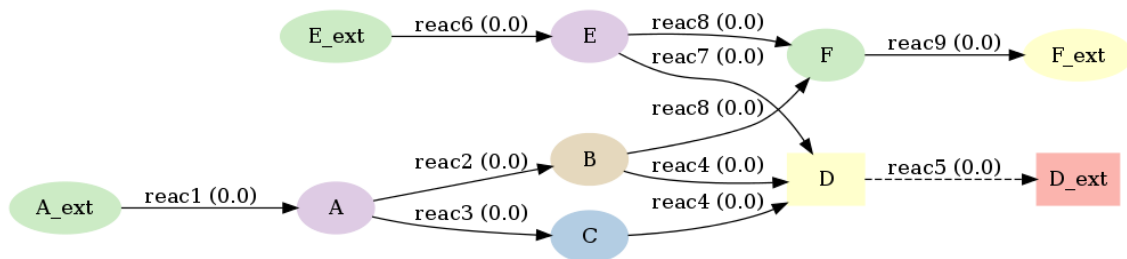


Figura 1. Representación del modelo juguete 1

Con el modelo juguete 1 se trató de hacer simulaciones previas en un sistema sencillo para observar el comportamiento del sistema en respuesta a pequeños cambios.

Este problema se abordó desde dos perspectivas, la primera cambiando los parámetros de entrada y salida, o dejándolos fijos y cambiando los parámetros internos. Sin embargo, hablamos de un modelo en el que todas sus reacciones se pueden poner en función de otra reacción (tabla 4) y al añadir la condición de estado estacionario, y las entradas que exige el algoritmo (al menos dos) y la salida de la función objetivo, los grados de libertad del modelo son cero. Por lo tanto, para cada conjunto de entradas y salidas sólo hay una solución óptima posible.

Nombre reacción : fórmula	
Condición estado estacionario:	$v_i = v_{pi} - v_{ci}$
Reac1:	$v_1 = x_1$
Reac2:	$v_2 = v_1 - v_3$
Reac3:	$v_3 = v_4$
Reac4:	$v_4 = v_3 = v_1 - v_2 = v_5 - v_7 = (v_1 - v_8)/2$
Reac5:	$v_5 = \text{crecimiento } (x_5)$
Reac6:	$v_6 = x_6$
Reac7:	$v_7 = v_5 - v_4$
Reac8:	$v_8 = (v_6 - v_7) = (v_1 - 2*v_4)$
Reac9:	$v_9 = 2*v_8$

Tabla 1. Relación matemática de los flujos del sistema donde  $v_i$  equivale al flujo de la reacción  $i$ ;  $x_i$  equivale a parámetros de entrada que aportas al sistema para ciertas reacciones  $i$ . Un FBA optimiza la reacción objetivo y los componentes que den lugar a ella y trata de evitar la formación de subproductos y la existencia de pérdidas. Como en este caso las entradas de la reacción 1 y la reacción 6 y la salida de la reacción 5 son fijas, la única pérdida posible del sistema la diferencia entre  $v_2$  y  $v_8$  restante que van a dar lugar a  $v_9$ , o lo que es lo mismo la diferencia entre el flujo de B restante y de E restante tras haber formado D, que van a dar lugar a la formación de F en una relación 1:2.

Al fijar las entradas  $x_1$  y  $x_6$ , y la salida  $x_5$ , la única distribución de flujos posible que cumple el estado estacionario obliga a que se produzca F, aunque se esté optimizando para la producción de D. Esto es porque la salida de F ( $x_9$ ) es el único grado de libertad que le queda al sistema tras fijar todo lo demás.

Esto se hace gráficamente visible en la figura 11 realizada con la plataforma online de cyanodesign, en la que los flujos se distribuyen de forma óptima por el sistema.

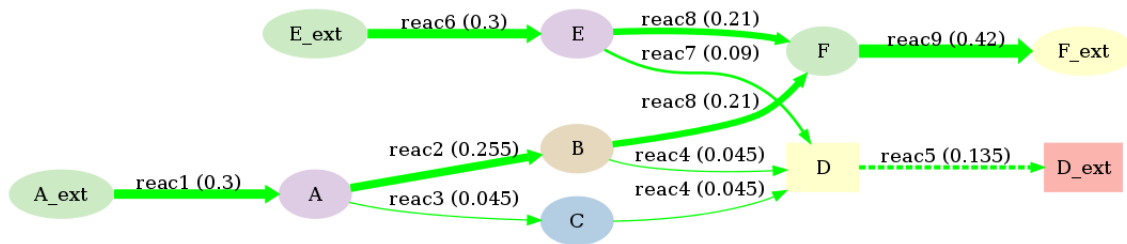


Figura 2. Distribución óptima de los flujos del modelo juguete 1, dadas las entradas  $v_1$  (A)=0.3,  $v_6$  (E)=0.3 y la salida de  $v_5$ , que es la reacción objetivo (equivalente al crecimiento en un sistema biológico),  $v_5$  (D)=0.135.

Para este modelo carece de sentido aplicar el algoritmo IOMA, que es el objeto de estudio en este proyecto, ya que al sólo existir una solución óptima el programa desestima todas las soluciones subóptimas que tienen en cuenta parámetros cinéticos ficticios. Por lo tanto para hacer comprobaciones simples del funcionamiento del programa es necesario contar con un modelo simple, pero que le dé cierta libertad al programa pero que permita varias soluciones óptimas para cada set de datos.

## 2. Modelo juguete 2

En este apartado de los anexos se presentan más ejemplos de las pruebas realizadas con el toy\_model2. En las siguientes figuras se observan tablas de datos correspondientes a los flujos de las reacciones. Presentan un código de colores en el que al azul son las entradas y salidas del sistema fijas y el resto de datos se representan con colores que van desde el rojo más intenso para las reacciones que presentan flujos negativos altos, rojo pálido para aquellas reacciones con valores negativos más cercanos o iguales a 0, naranja claro para reacciones de valores positivos cercanos a 0, verde pálido para reacciones con flujos positivos iguales a uno, verde intenso para aquellas reacciones con flujos positivos superiores a uno, y verde oscuro para las reacciones que presentan los valores de flujo más altos del sistema.

En la figura 3 se puede observar que el FBA y el IOMA toman caminos distintos para llegar a un mismo flujo debido a que el algoritmo IOMA tiene en consideración los parámetros proteómicos y metabolómicos que se le aportan. Sin embargo, es consistente en sus simulaciones ya que, con los mismos datos proteómicos y metabolómicos y diferente flujo exigido en su reacción objetivo, el algoritmo IOMA sigue tomando en todos los casos el mismo camino para alcanzar este flujo. Como se menciona en el texto la reacción objetivo es reac5,



cuyo flujo es un dato necesario del algoritmo. En este caso los datos proteómicos introducidos son ficticios y pretendían minimizar el flujo en la reacción 3 y maximizar el flujo en la reacción 11, ambas pretensiones se cumplen, como se puede ver en la figura 3. Además se pretendía que la reacción 16 tuviera flujos crecientes en las distintas condiciones (bajos en la condición 1, medios en la condición 2, y altos en la condición 3). Esta última condición no se cumple ya que todos los flujos de esta reacción son nulos. Esto puede ser debido a que el programa no encontraba una solución óptima en la que esta reacción tuviera flujo o a que la entrada del metabolito precursor de la 'ruta' en la que se encuentra esta reacción no es una entrada fija del sistema y el algoritmo prefiere tratar de utilizar los nutrientes cuya entrada es fija para la producción del objetivo.

	FBA-condición1	IOMA-condición1	FBA-cond2	IOMA-cond2	FBA-cond3	IOMA-cond3
reac1	1	1	1	1	1	1
reac2	0,952	1	0,96	1	0,92	1
reac3	0,048	0	0,04	0	0,08	0
reac4	0,048	0	0,04	0	0,08	0
reac5	0,145	0,145	0,12	0,12	0,25	0,250
reac6	1	1	1	1	1	1
reac7	0,097	0	0,08	0	0,17	0
reac8	0,903	1	0,92	1	0,83	1
reac9	1,807	2	1,84	2	1,67	2
reac10	0	0	0	0	0	0
reac11	0	0,145	0	0,12	0	0,250
reac12	0	0	0	0	0	0
reac13	0	0	0	0	0	0
reac14	0	0	0	0	0	0
reac15	0	0	0	0	0	0
reac16	0	0	0	0	0	0
reac17	0	0	0	0	0	0
reac18	0	0	0	0	0	0
reac19	0	0,145	0	0,12	0	0,250
reac20	0	0,145	0	0,12	0	0,250
A_ext_trans	1	1	1	1	1	1
E_ext_trans	1	1	1	1	1	1
D_ext_trans	-0,145	-0,145	-0,12	-0,12	-0,25	-0,250
H_ext_trans	0	0	0	0	0	0
G_ext_trans	0	0	0	0	0	0
F_ext_trans	-1,807	-2	-1,84	-2	-1,67	-2
K_ext_trans	0	0,145	0	0,12	0	0,250

Figura 3. Simulación realizada con el toy\_model2

En la figura 4 se puede observar otro ejemplo de las simulaciones realizadas con este modelo. En el caso de las figuras 4 y 5 solo encontramos un resultado para el FBA porque solo se utilizan unas condiciones de entrada y de salida de nutrientes. Se mantienen fijas en ambos casos las reacciones 1, 5,6 y 19 que corresponden a las entradas de A, E y K; y a la salida de D.

En ambas simulaciones se incorporan datos proteómicos de las reacciones 4 y 17. Además en la simulación representada en la figura 4 también se condicionó la reacción 7, cuyos datos indicaban que en la condición 2 el flujo era alto, en la condición 3 el flujo era medio y en las condiciones 1 y 4 era bajo. En esta misma simulación se utilizaron datos ficticios con la intención de que en la condición 1 y 2, la reacción 4 tuviera flujos bajos; que estos fueran

medios para la condición 4, y altos para la condición 3. La reacción 17 se pretendía que los flujos más altos estuvieran en las condiciones 1 y 2, en la condición 3 el flujo fuera bajo y en la 4 fuera medio. A grandes rasgos podemos observar que cumplen las tendencias y arrojan resultados más realistas ya que distribuyen flujo por aquellas reacciones que tienen flujo según los datos proteómicos.

	FBA	IOMA1	IOMA2	IOMA3	IOMA4
reac1	3	3	3	3	3
reac2	1,5	3	2,896	2,208	2,625
reac3	1,5	0	0,104	0,792	0,375
reac4	1,5	0	0,104	0,792	0,375
reac5	5	5	5	5	5
reac6	3	3	3	3	3
reac7	2	0	0,208	0,083	0
reac8	0	3	2,792	1,417	2,25
reac9	1	6	5,583	4,333	5,25
reac10	0	0	0	0	0
reac11	0	0	2,438	3	2,938
reac12	3	3	0,563	0	0,063
reac13	3	3	3	3	3
reac14	0,5	1,143	0,75	0,375	0,563
reac15	2,5	1,857	2,25	2,625	2,438
reac16	0	1,571	0	0	0
reac17	1,5	1,857	2,25	1,125	1,688
reac18	1	0	0	1,5	0,75
reac19	3	3	3	3	3
reac20	3	3	3	3	3

Figura 4. Simulación realizada con el toy\_model2 en diferentes condiciones.

En la figura 5 encontramos los resultados de otra simulación con los mismos datos de entrada y de salida que la simulación anterior. Las reacciones limitadas son la 4, la 14 y la 17. De igual forma que en la figura anterior se cumplen las tendencias. En la simulación cuyos resultados se encuentran en la figura 5, se introdujeron datos elaborados manualmente con el fin de que la reacción 4 tuviera flujos altos en la condición 2, medios en las condiciones 1 y 4, y bajos en la condición 3. La reacción 17 se trató de que tuviera resultados con flujos altos en las dos primeras condiciones y bajos en las dos restantes. A la reacción 14 se le incorporaron datos 'proteómicos' y 'metabolómicos' con la intención de que esta tuviera un flujo alto en la tercera condición, medios en la primera y segunda condición y bajos en la cuarta condición.

	FBA	IOMA1	IOMA2	IOMA3	IOMA4
reac1	3	3	3	3	3
reac2	1,5	2,043	1,5	2,418	2,056
reac3	1,5	0,957	1,5	0,582	0,944
reac4	1,5	0,957	1,5	0,582	0,944
reac5	5	5	5	5	5
reac6	3	3	3	3	3
reac7	2	1,793	1,25	0	0,127
reac8	0	1,086	0	1,836	1,111
reac9	1	2,172	0	4,836	3,984
reac10	0	0,121	1,75	0	0
reac11	0	0	0	0	3
reac12	3	3,121	4,75	3	0
reac13	3	3	3	3	3
reac14	0,5	0,75	0,75	0,893	0,310
reac15	2,5	2,25	2,25	2,107	2,691
reac16	0	0	0	1,738	0
reac17	1,5	2,25	2,25	0,943	0,929
reac18	1	0	0	1,164	1,762
reac19	3	3	3	3	3
reac20	3	3	3	3	3

Figura 5. Simulación realizada con el toy\_model2



## ANEXO III

2.7.7.27	alpha-D-glucose 1-phosphate + ATP -> ADP-D-glucose + diphosphate
_a-glucansyn	2 ADP-D-glucose -> a 1,4-alpha-D-glucan_n
_2.4.1.21a	ADP-D-glucose + a 1,4-alpha-D-glucan_n -> ADP + a 1,4-alpha-D-glucan_n1
_glycsyn	2 ADP-D-glucose + a 1,4-alpha-D-glucan_n1 -> a glycogen_n
_2.4.1.21b	ADP-D-glucose + a glycogen_n -> ADP + a glycogen_n1
_2.4.1.1	a glycogen_n1 + phosphate O4P -> a glycogen_n + alpha-D-glucose 1-phosphate
_2.4.1.18	a glycogen_n1 + a 1,4-alpha-D-glucan_n <-> a glycogen_n + a 1,4-alpha-D-glucan_n1
Glycbm	a glycogen_n1 -> Glycbm
glycout	a glycogen_n1 -> a glycogen_n1_out

Tabla 1. Reacciones en las que interviene el glucógeno o sus precursores.

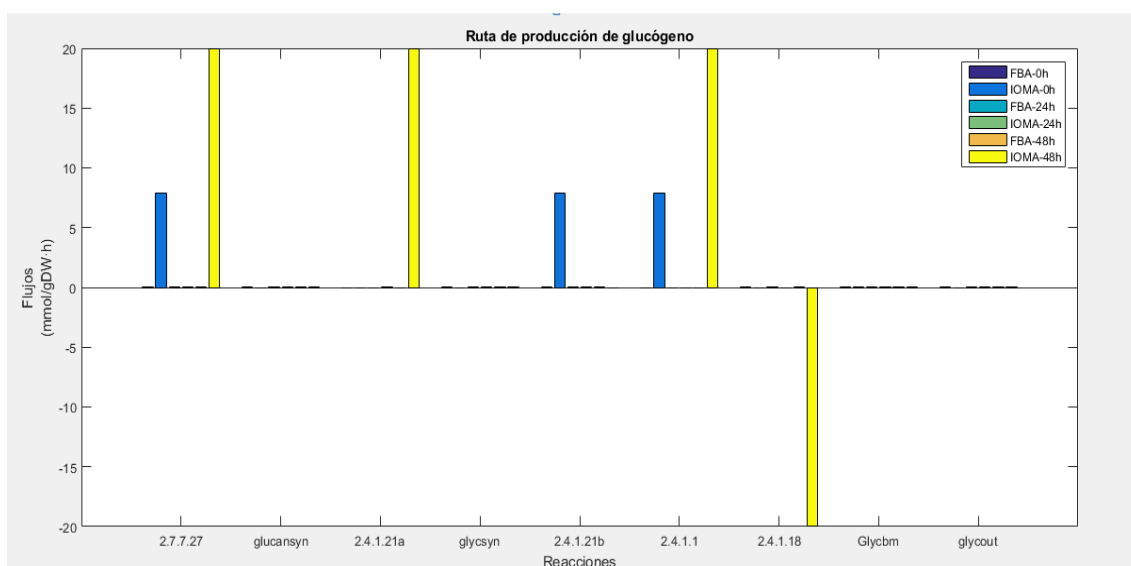


Figura 1. Flujo en las reacciones en las que interviene el glucógeno o sus precursores. Se pueden observar dos ciclos fútiles en los resultados aportados por el algoritmo IOMA en las condiciones de presencia de nitrógeno y 48 horas en ausencia de nitrógeno.

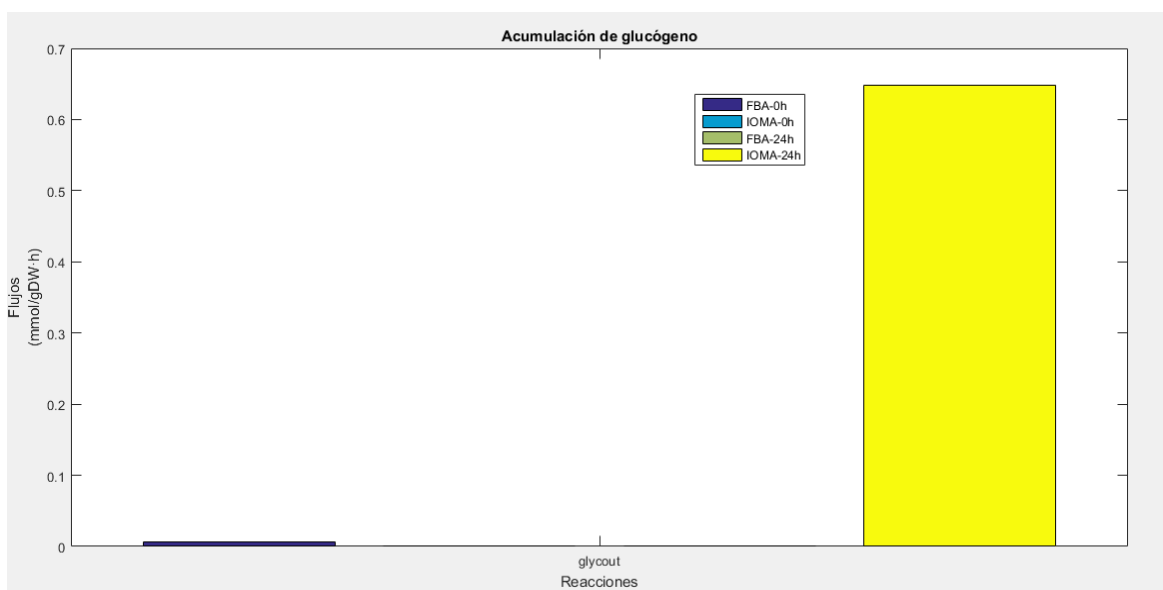


Figura 2. Resultados arrojados por los dos métodos de análisis. Flujo en la reacción de externalización del nitrógeno en la simulación de la condición con nitrógeno y tras 24 horas sin nitrógeno.

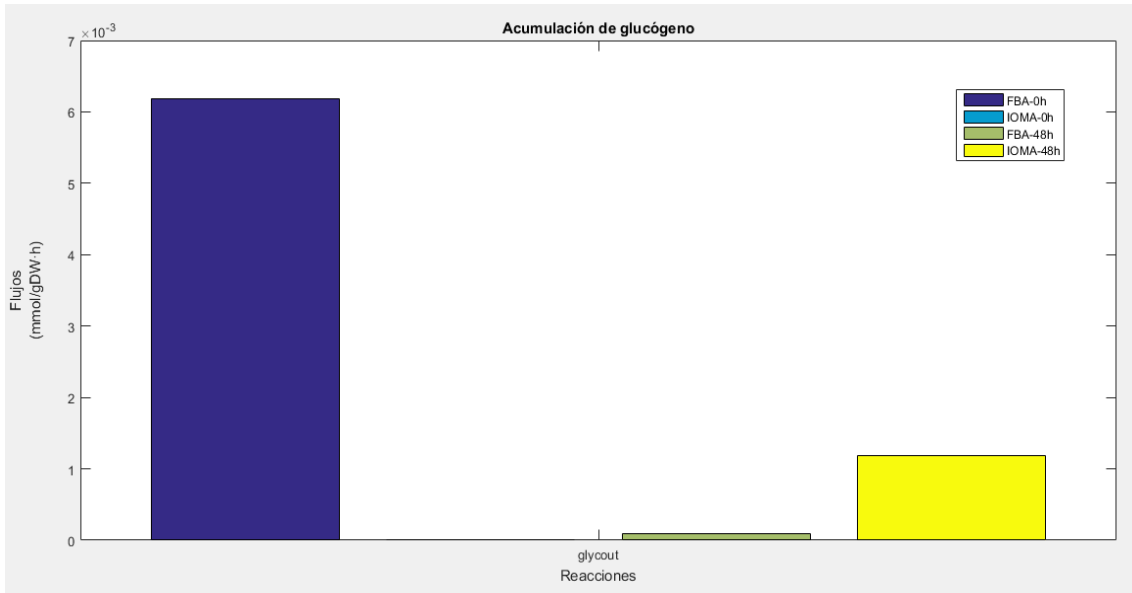


Figura 3. Resultados de ambos algoritmos para la acumulación de nitrógeno. Las condiciones simuladas son de presencia de nitrógeno y 48 horas de ausencia de nitrógeno.

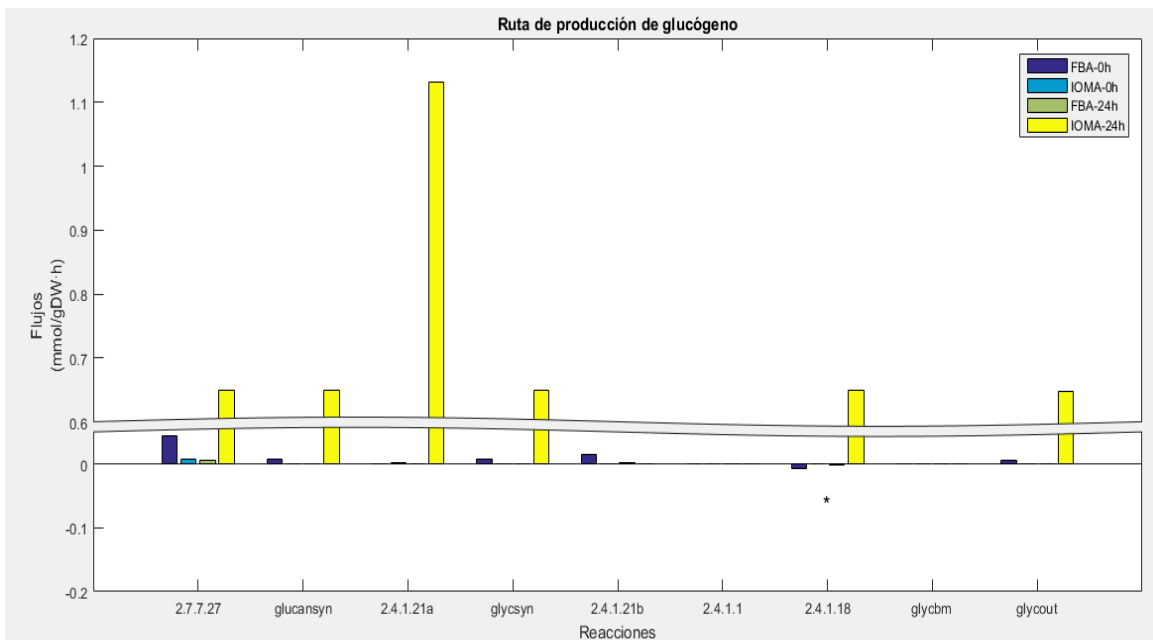


Figura 4. Flujos en la ruta de producción de nitrógeno arrojados por ambos algoritmos. Las condiciones comparadas son presencia de nitrógeno y 24 horas de ausencia de nitrógeno.

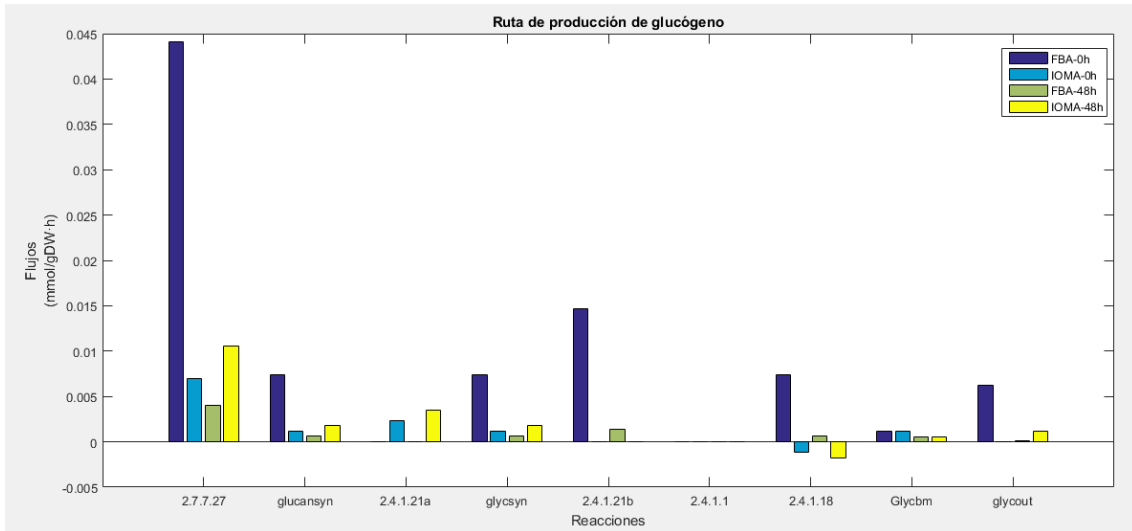


Figura 5. Resultados de las reacciones de producción de glucógeno o sus precursores. Resultados aportados por ambos algoritmos en presencia de nitrógeno y tras 48 horas en ausencia de nitrógeno.

6.2.1.5	succinate + coenzyme A + ATP <-> succinyl-CoA + ADP + phosphate O4P
4.2.1.2	fumarate + H2O <-> malate
1.1.1.37	malate + NAD+ <-> oxaloacetate + NADH + H+
2.3.3.1	oxaloacetate + acetyl-CoA + H2O <-> citrate + coenzyme A
4.2.1.3a	citrate <-> cis-aconitate + H2O
4.2.1.3b	cis-aconitate + H2O <-> D-isocitrate
1.1.1.42	D-isocitrate + NADP+ <-> 2-ketoglutarate + CO2 + NADPH + H+

Tabla 2. Ciclo del ácido cítrico.

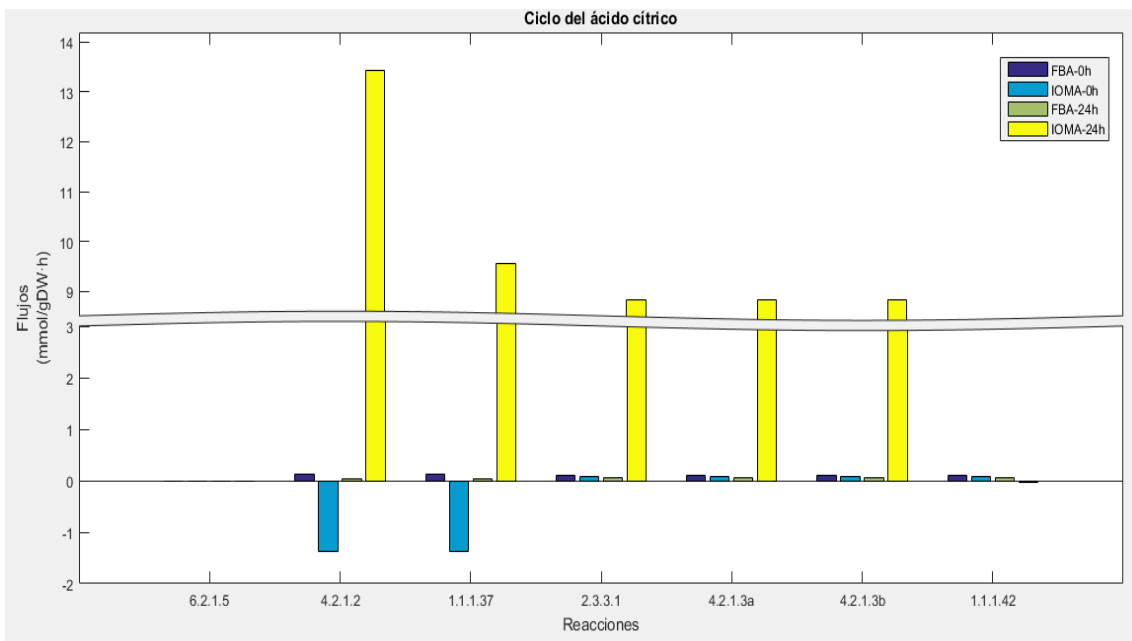


Figura 6. Resultados de flujos del ciclo del ácido cítrico calculados con el FBA y con el IOMA para dos condiciones. Las dos condiciones son presencia de nitrógeno y 24 horas de ausencia de nitrógeno.

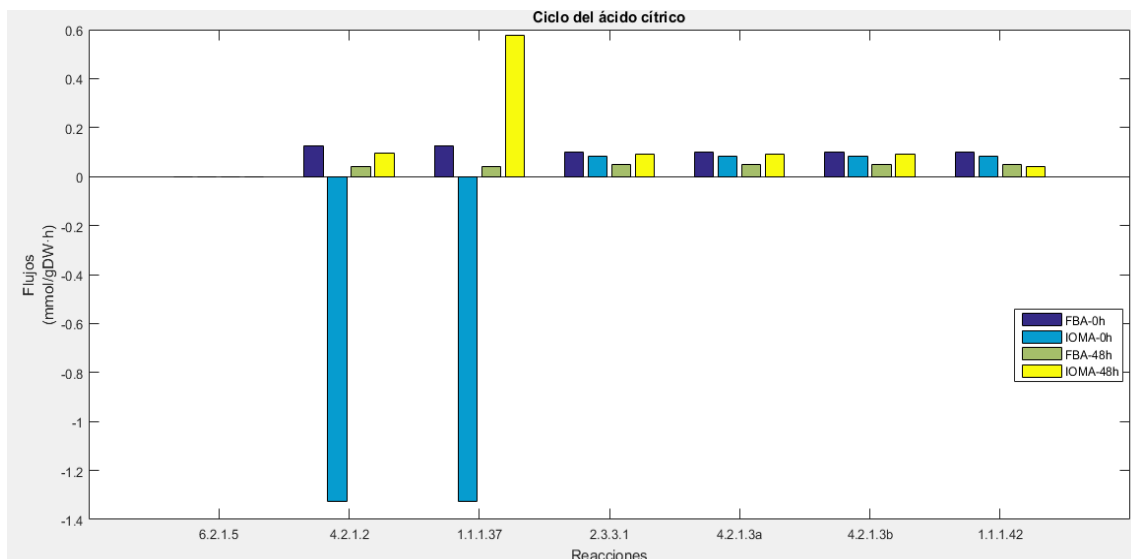


Figura 7. Figura 6. Resultados de flujos del ciclo del ácido cítrico calculados con el FBA y con el IOMA para dos condiciones. Las dos condiciones son presencia de nitrógeno y 48 horas de ausencia de nitrógeno.

5.3.1.9a	alpha-D-glucose 6-phosphate <-> beta-D-glucose-6-phosphate
5.3.1.9b	beta-D-glucose-6-phosphate <-> fructose-6-phosphate
5.3.1.9c	alpha-D-glucose 6-phosphate <-> fructose-6-phosphate
5.4.2.2	alpha-D-glucose 1-phosphate <-> alpha-D-glucose 6-phosphate
2.7.1.11	ATP + fructose-6-phosphate -> ADP + fructose-1,6-bisphosphate
3.1.3.11	fructose-1,6-bisphosphate + H2O -> fructose-6-phosphate + phosphate O4P
4.1.2.13a	fructose-1,6-bisphosphate <-> dihydroxy-acetone phosphate + D-glyceraldehyde-3-phosphate
5.3.1.1	D-glyceraldehyde-3-phosphate <-> dihydroxy-acetone phosphate
1.2.1.12	D-glyceraldehyde-3-phosphate + phosphate O4P + NAD+ + H+ <-> 1,3-diphosphateglycerate + NADH
2.7.2.3	3-phosphoglycerate + ATP <-> 1,3-diphosphateglycerate + ADP
5.4.2.12	3-phosphoglycerate <-> 2-phosphoglycerate
4.2.1.11	2-phosphoglycerate <-> phosphoenolpyruvate + H2O
2.7.1.40a	ADP + phosphoenolpyruvate -> ATP + pyruvate
2.7.1.40b	dADP + phosphoenolpyruvate <-> dATP + pyruvate
2.7.1.40c	GDP + phosphoenolpyruvate <-> GTP + pyruvate
2.7.1.40d	dGDP + phosphoenolpyruvate <-> dGTP + pyruvate
1.2.4.1	pyruvate + lipoate acetyltransferase N6-(lipoyl)lysine <-> lipoate acetyltransferase N6-(S-acetyldihydrolipoyl)lysine + CO2
2.3.1.12	acetyl-CoA + lipoate acetyltransferase N6-(dihydrolipoyl)lysine <-> coenzyme A + lipoate acetyltransferase N6-(S-acetyldihydrolipoyl)lysine
6.2.1.1	coenzyme A + acetate + ATP <-> acetyl-CoA + diphosphate + AMP
1.2.1.3c	acetaldehyde + NAD+ + H2O <-> NADH + acetate + H+
1.1.1.1	acetaldehyde + NADH + H+ -> ethanol + NAD+
1.1.1.2	NADPH + acetaldehyde + H+ -> NADP+ + ethanol

Tabla 3. Reacciones utilizadas para evaluar la ruta de la glicolisis.

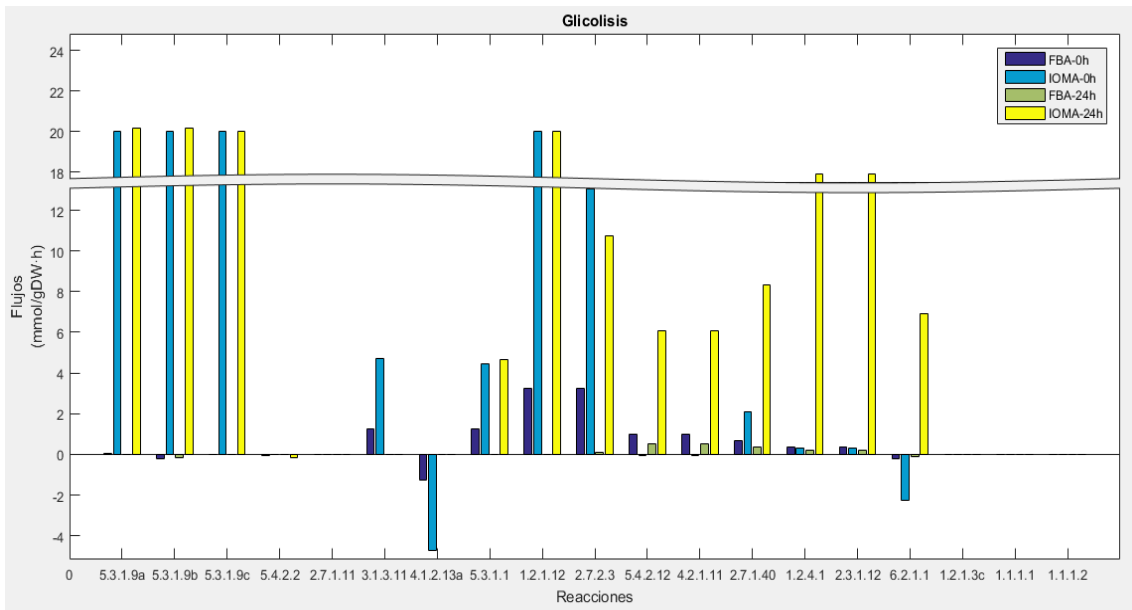


Figura 8. Resultados obtenidos en la ruta de la glicolisis en 2 condiciones, presencia de nitrógeno y 24 horas en ausencia de nitrógeno.

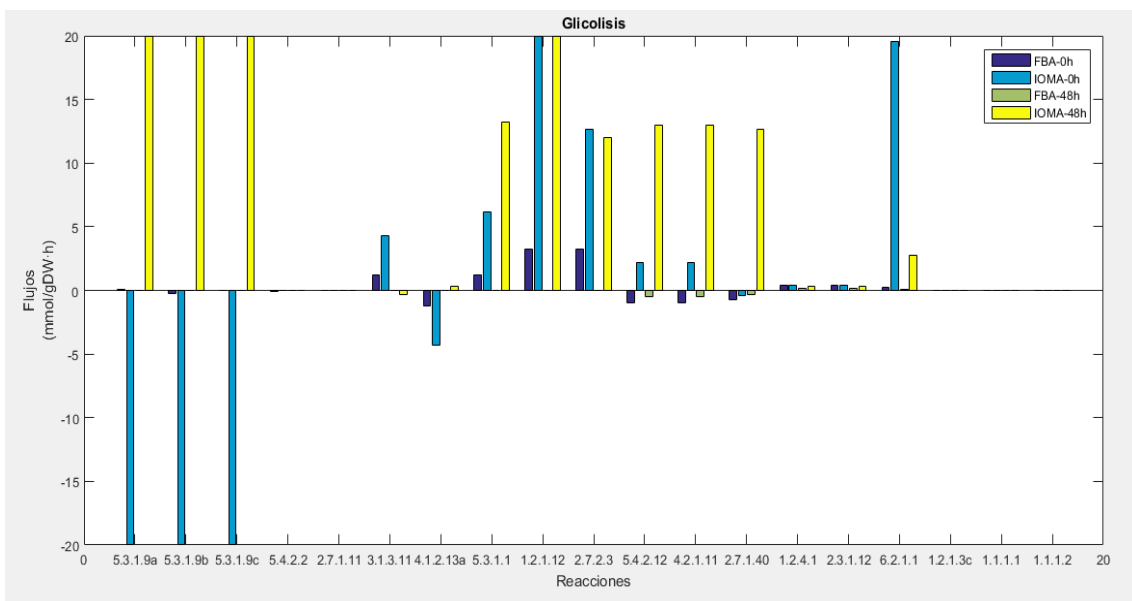


Figura 9. Resultados obtenidos en la ruta de la glicolisis en 2 condiciones, presencia de nitrógeno y 48 horas en ausencia de nitrógeno.

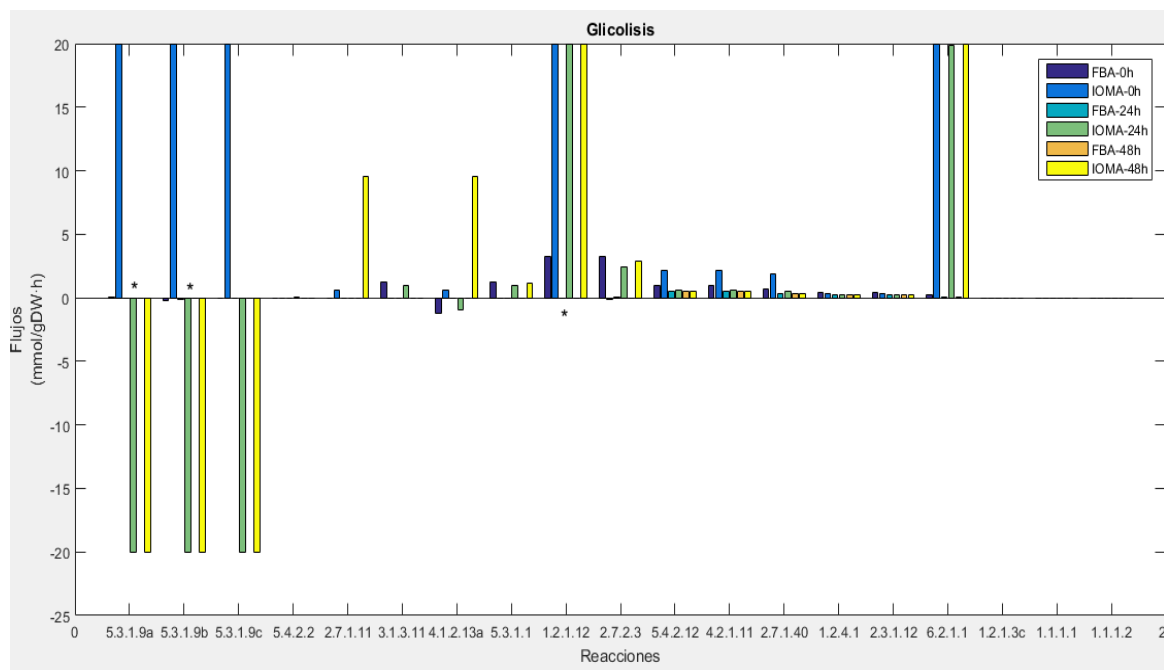


Figura 10. Resultados obtenidos en la ruta de la glicolisis en las 3 condiciones teniendo en cuenta las tres primeras reacciones que son las de interconversión entre glucosas, y conversión de estas a fructosa.

4.1.1.31	phosphoenolpyruvate + H <sub>2</sub> O + CO <sub>2</sub> -> phosphate O4P + oxaloacetate
1.1.1.37	oxaloacetate + NADH + H <sup>+</sup> <-> malate + NAD <sup>+</sup>
1.1.1.38	NAD <sup>+</sup> + malate -> NADH + CO <sub>2</sub> + pyruvate + H <sup>+</sup>
2.7.1.40a	ADP + phosphoenolpyruvate -> ATP + pyruvate
2.7.1.40b	dATP + pyruvate <-> dADP + phosphoenolpyruvate
2.7.1.40c	GTP + pyruvate <-> GDP + phosphoenolpyruvate
2.7.1.40d	dGTP + pyruvate <-> dGDP + phosphoenolpyruvate
4.2.3.3	dihydroxy-acetone phosphate -> methylglyoxal + phosphate O4P
4.4.1.5	methylglyoxal + glutathione <-> S-lactoyl-glutathione
3.1.2.6	S-lactoyl-glutathione + H <sub>2</sub> O -> glutathione + D-lactate
1.1.1.28	NAD <sup>+</sup> + D-lactate -> NADH + pyruvate + H <sup>+</sup>

Tabla 4. Reacciones utilizadas para analizar el metabolismo del piruvato.



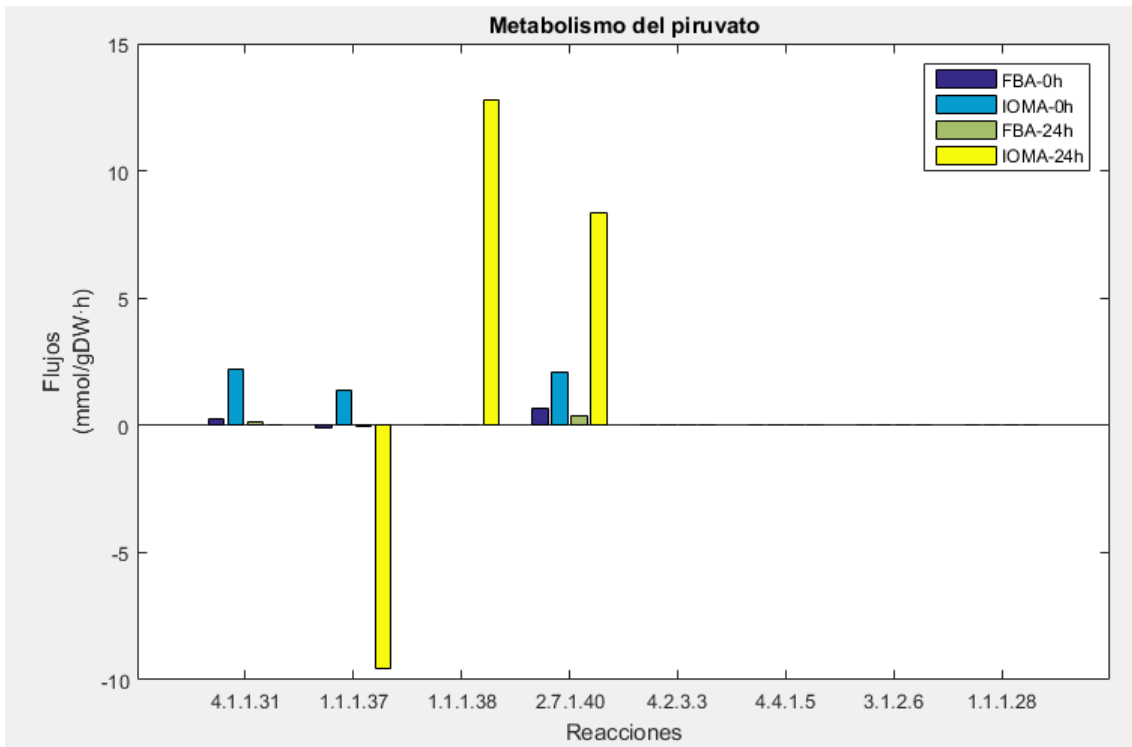


Figura 11. Figura 12. Resultados obtenidos las reacciones del metabolismo del piruvato en 2 condiciones. En esta figura se representan los resultados que arrojan los dos algoritmos en condiciones de presencia de nitrógeno, y 24 horas en ausencia de nitrógeno.

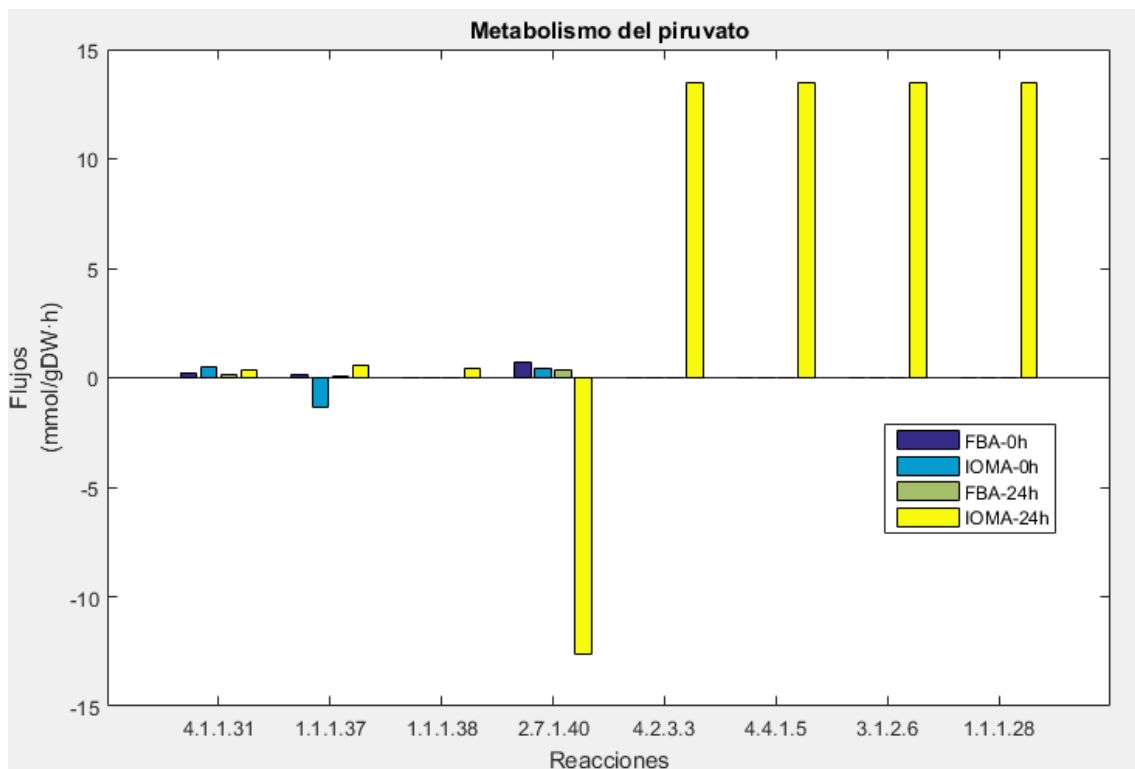


Figura 12. Resultados obtenidos las reacciones del metabolismo del piruvato en 2 condiciones. En esta figura se representan los resultados que arrojan los dos algoritmos en condiciones de presencia de nitrógeno, y 48 horas en ausencia de nitrógeno.

1.1.1.47a	beta-D-glucose + NAD+ -> D-glucono-1,5-lactone + NADH + H+
1.1.1.47b	beta-D-glucose + NADP+ -> D-glucono-1,5-lactone + NADPH + H+
1.1.5.2	beta-D-glucose + UQ -> D-glucono-1,5-lactone + UQH2
4.1.2.14	2-keto-3-deoxy-6-phospho-gluconate -> D-glyceraldehyde-3-phosphate + pyruvate
3.1.1.31	D-glucono-delta-lactone-6-phosphate + H2O -> 6-phospho-D-gluconate
1.1.1.44	6-phospho-D-gluconate + NADP+ -> D-ribulose-5-phosphate + CO2 + NADPH + H+
1.1.1.49	beta-D-glucose-6-phosphate + NADP+ <-> D-glucono-delta-lactone-6-phosphate + NADPH + H+
5.3.1.9a	alpha-D-glucose 6-phosphate <-> beta-D-glucose-6-phosphate
5.3.1.9c	alpha-D-glucose 6-phosphate <-> fructose-6-phosphate
2.7.1.11	ATP + fructose-6-phosphate -> ADP + fructose-1,6-bisphosphate
4.1.2.13a	fructose-1,6-bisphosphate <-> dihydroxy-acetone phosphate + D-glyceraldehyde-3-phosphate
4.1.2.4	deoxyribose-5-phosphate <-> acetaldehyde + D-glyceraldehyde-3-phosphate
2.2.1.1a	D-ribose-5-phosphate + D-xylulose-5-phosphate <-> D-sedoheptulose-7-phosphate + D-glyceraldehyde-3-phosphate
2.2.1.1b	D-erythrose-4-phosphate + D-xylulose-5-phosphate <-> fructose-6-phosphate + D-glyceraldehyde-3-phosphate
2.2.1.2	D-glyceraldehyde-3-phosphate + D-sedoheptulose-7-phosphate <-> fructose-6-phosphate + D-erythrose-4-phosphate
5.1.3.1	D-ribulose-5-phosphate <-> D-xylulose-5-phosphate
5.3.1.6	D-ribose-5-phosphate <-> D-ribulose-5-phosphate
2.7.6.1	ATP + D-ribose-5-phosphate -> 5-phosphoribosyl 1-pyrophosphate + AMP

Tabla 5. Reacciones utilizadas para analizar la ruta de las pentosas fosfato.

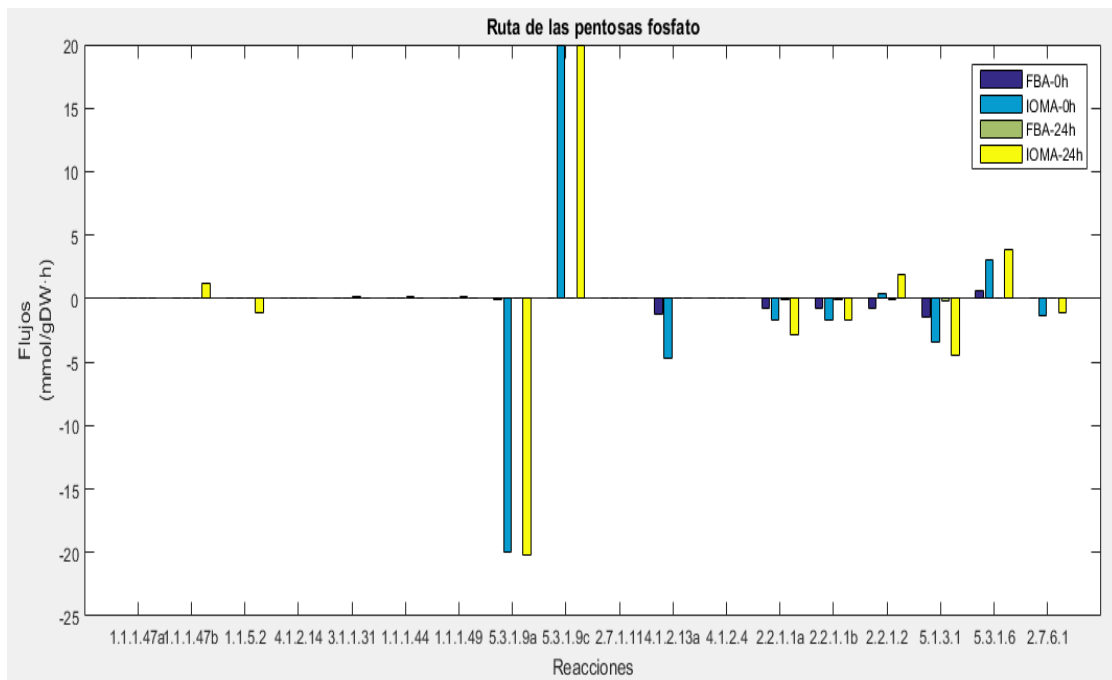


Figura 13. Resultados obtenidos utilizando ambos algoritmos en la ruta de las pentosas fosfato. En esta simulación se utilizaron datos de dos condiciones, presencia de nitrógeno y 48 horas en ausencia de nitrógeno

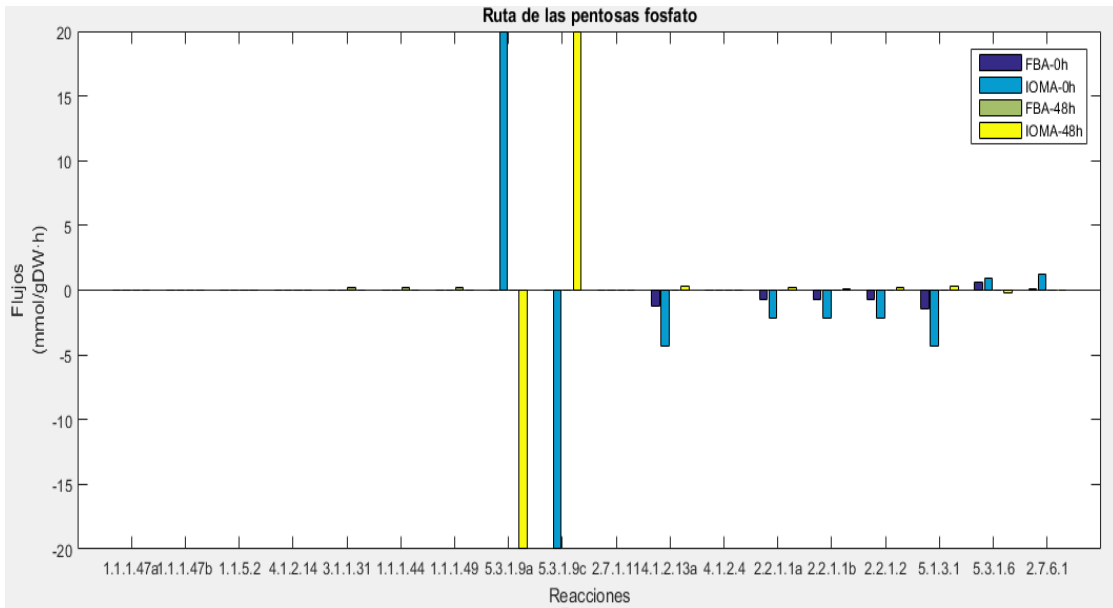


Figura 14. Flujos obtenidos utilizando ambos algoritmos en la ruta de las pentosas fosfato. En esta simulación se utilizaron datos de dos condiciones, presencia de nitrógeno y 48 horas en ausencia de nitrógeno.