Table of Contents

Supplementary Methods

1. One-Step pFBA

Parsimonious FBA (pFBA) finds a flux distribution (v) which maximizes an objective function (e.g., biomass or metabolite production) and also minimizes the total flux through the network. pFBA can be formulated as two optimization problems (e.g., maximize biomass and then minimize the sum of the absolute values of fluxes). In this work, a single optimization problem was formulated to maximize production of a metabolite and minimize the sum of the absolute values of fluxes:

	$\max \sum_{i \in MoI} v_i^{sink} - \sum_{j \in J} \epsilon \cdot (v_j^{forward} + v_j^{reverse})$	(S1)
	$\sum_{j \in J} S_{ij} \cdot v_j - v_i^{sink} = 0, \quad \forall i \in MoI$	(S2)
	$\sum_{j \in J} S_{ij} \cdot v_j = 0, \quad \forall i \notin MoI$	(S3)
	$v_j = v_j^{forward} - v_j^{reverse}, \quad \forall j \in J$	(S4)
	$v_j^{Lower} \leq v_j \leq v_j^{Upper} \quad \forall j \in J$	(S5)

Here v_i^{sink} is a non-negative flux variable that removes a metabolite of interest (MoI) from the network, $v_j^{forward}$ and $v_j^{reverse}$ are non-negative flux variables used to represent the net flux (v_j) through a reaction j, (**Eq. S4**). J is the set of all reactions. The upper and lower limits for each flux are v_j^{Lower} and v_j^{Upper}, respectively. ϵ is a parameter used to penalize total flux usage and, unless stated otherwise, was set to 10^{-3}. Alternatively, to optimize for production of biomass **Eq. S3-S5**, would be imposed as constraints and **Eq. S6** used as an objective function. Since $v_{Biomass}$ is much smaller than the total fluxes a smaller value for ϵ was used ($\epsilon = 1e-8$) in **Eq. S6**.

	$\max \ v_{Biomass} - \sum_{j \in J} \epsilon \cdot (v_j^{forward} + v_j^{reverse})$	(S6)

2. Finding Alternate MapMaker and PathTracer Solutions Using Integer Cuts

2.1 MapMaker Integer Cuts:

To generate alternate MapMaker solutions that have different CTMs (i.e., different γ_{Crp}), integer constraints (**Eq. S7**) can be added using solutions for γ_{Crp} (which indicates whether C is transferred from r to p) from each previous solution, k:

	$\sum_{(r,p) \in (R,P)} \gamma_{Crp} \cdot \gamma_{Crp}^k \leq \sum_{(r,p) \in (R,P)} \gamma_{Crp}^k - 1, \quad \forall k = 1, 2, \dots K-1$	(S7)

Here K-1 is the total number of previous solutions and γ_{Crp}^k are the corresponding solutions for γ_{Crp} at each previous solution, k.

2.2 PathTracer Integer Cuts:

Two types of integer cuts can be used to find different types of alternate PathTracer solutions.

2.2.1 Find Another Path Missing at Least One Edge from Each Prior Path: The first type of cuts guarantee that any new path proposed does not use at least one edge from a prior path and is formulated as:

	$\sum_{(i,j,n) \in Map_{nji}} \sum_{d=1}^{|D|} a_{njid} \cdot StoreRxn_j^k + \sum_{(i,j,n) \in Map_{ijn}} \sum_{d=1}^{|D|} a_{njid}^{rev} \cdot StoreRxn_j^k \leq \sum_j StoreRxn_j^k - 1, \forall k = 1, 2, \dots K-1$	(S8)

Here $StoreRxn_j^k$ is a binary parameter that is set to 1 if a reaction, j, was part of a path in the prior solution k. In this case, a new integer cut (**Eq. S8**) is added each time a new path is discovered.

2.2.2 Find Another Path that Uses a Previously Unused Reaction: The second type of cut requires that a new path use at least one edge that was not used in any previous paths. Only one integer cut constraint is needed (**Eq. S9**).

$$\sum_{(i,j,n)\in Map_{nji}} \sum_{d=1}^{|D|} a_{njid} \cdot (1 - StorePath_{nji}) + \sum_{(i,j,n)\in Map_{ijn}} \sum_{d=1}^{|D|} a_{njid}^{rev} \cdot (1 - StorePath_{nji}) \geq 1 \quad \text{(S9)}$$

Here $StorePath_{nji}$ is a binary parameter that is set to one if *any* prior path uses reaction, j, to connect metabolite, n, to metabolite, i. This approach does not increase the number of integer cuts as additional solutions are generated; instead, the set of $StorePath_{nji}$ parameters change with each new solution. This approach will not propose every solution, since using a reaction that is functionally equivalent to a reaction used in previous paths (e.g. different transport mechanism or different electron carrier) will only count as a new path the first time it is used.

3. Bilevel Program to Eliminate All Paths Between Two Metabolites

In the main text, PathTracer enumerated all paths between two metabolites—putrescine and glutamate. These paths were subsequently evaluated to identify reaction deletions that eliminate these paths (in this case, creating essentiality conditions). A more direct approach for finding reaction deletions that eliminate all paths between two metabolites uses bilevel optimization, which includes an inner problem and an outer problem. The inner problem determines whether, when subjected to a set of deletions (provided by the outer problem), there remains a path between the starting and ending metabolite. The presence or absence of a path is then used by the outer problem to find a set of deletions that eliminate all paths.

3.1 Inner Problem:

The inner problem uses the following objective function (**Eq. 10**) and constraints (**Eq. S11-S15**):

$$\max \sum_{n\in EndNode} Sink_n \quad \text{(S10)}$$

The inner problem is similar to PathTracer, except that depth information is omitted and all flow variables ($Sink$, $Source$, a and a^{rev}) are non-negative variables rather than binary variables. These changes allow the inner optimization problem to be converted into a set of linear constraints using primal-dual equality. **Eq. S10** maximizes flow though the sink variable ($Sink_q$) by finding a path from the start to the end node(s). The paths available are constrained by the set of maps (e.g., CTMs) and flow balances over each node (**Eq. S11-S13**):

$$\sum_{(i,j)\in Map_{ijn}} (a_{ijn} - a_{nji}^{rev}) + \sum_{(i,j)\in Map_{nji}} (a_{ijn}^{rev} - a_{nji}) + Source_n = 0, \qquad n = StartNode \quad \text{(S11)}$$

$$\sum_{(i,j)\in Map_{ijn}} (a_{ijn} - a_{nji}^{rev}) + \sum_{(i,j)\in Map_{nji}} (a_{ijn}^{rev} - a_{nji}) - Sink_n = 0, \qquad n = EndNode \quad \text{(S12)}$$

$$\sum_{(i,j)\in Map_{ijn}} (a_{ijn} - a_{nji}^{rev}) + \sum_{(i,j)\in Map_{nji}} (a_{ijn}^{rev} - a_{nji}) = 0, \qquad \forall n | n \notin (StartNode \cup EndNode) \quad \text{(S13)}$$

The node balance is similar to the PathTracer's except that depth information is removed. (Note for loops an additional constraint is needed that requires the sum of a and a^{rev} over the set *Map* to be greater or equal to the sum of all source flows. Such a constraint prevents a source edge from feeding directly to the sink on the same node, but also allows for the inner problem to still determine whether a path exists.). To capture outer problem deletions in the inner problem, the following constraint is used:

$$\sum_{(n,i)\in Map_{nji}} a_{nji} + \sum_{(n,i)\in Map_{ijn}} a_{nji}^{rev} \leq \beta_j, \qquad \forall j \in J \quad \text{(S14)}$$

This is the same reaction removal constraint used in PathTracer except the binary variable, β_j, is used to allow the outer problem to select reaction deletions. To eliminate particular nodes (e.g., CoA and ACP) from being used in a path (**Eq. S15**) can be used.

$$\sum_{(i,j)\in Map_{nji}} a_{nji} + \sum_{(i,j)\in Map_{ijn}} a_{nji}^{rev} \leq 0, \qquad \forall n | n = Eliminated\ Node \quad \text{(S15)}$$

The inner optimization problem is replaced by a set of linear constraints formulated from the primal (**Eq. S10-S15**) and the corresponding dual (not shown) problems.

3.2 Outer Problem to Eliminate All Paths:

The inner problem tries to find a path subject to the reaction deletions chosen by the outer problem. To find a minimal set of reaction deletions that eliminates all paths between two metabolites then outer problem would decide values for the non-negative variables, β_j by minimizing the following outer objective function (**Eq. S16**):

$$\min \sum_{n \in EndNode} Sink_n + 0.001 \cdot \sum_{j \in J} (1 - \beta_j) \quad \text{(S16)}$$

3.3 Outer Problem to Eliminate All Paths Except Those Using a Particular Reaction:

The FOCAL algorithm identifies essentiality conditions for a given reaction of interest (RoI), where the conditions involve media and gene deletions so that growth can occur if the RoI takes place and growth can not occur if the RoI does not take place. A similar problem can be formulated as a bilevel pathfinding problem, where paths between a carbon source and a biomass component are all eliminated except paths involving the RoI. In this case, the outer problem chooses a set of reactions such that no path exists in the inner problem when the reaction of interest (RoI) and the set of reactions are removed, but when RoI remains in the network a path still exists. The outer problem is formulated using **Equations S17-S30**:

$$\sum_{(i,j) \in Map_{ijn}} (A_{ijn} - A^{rev}_{nji}) + \sum_{(i,j) \in Map_{nji}} (A^{rev}_{ijn} - A_{nji}) + SOURCE_n = 0, \qquad n = StartNode \quad \text{(S17)}$$

$$\sum_{(i,j) \in Map_{ijn}} (A_{ijn} - A^{rev}_{nji}) + \sum_{(i,j) \in Map_{nji}} (A^{rev}_{ijn} - A_{nji}) - SINK_n = 0, \qquad n = EndNode \quad \text{(S18)}$$

$$\sum_{(i,j) \in Map_{ijn}} (A_{ijn} - A^{rev}_{nji}) + \sum_{(i,j) \in Map_{nji}} (A^{rev}_{ijn} - A_{nji}) = 0, \qquad \forall n | n \notin (StartNode \cup EndNode) \quad \text{(S19)}$$

$$\sum_{(n,i) \in Map_{nji}} A_{nji} + \sum_{(n,i) \in Map_{ijn}} A^{rev}_{nji} \leq 1, \qquad \forall j \in RoI \quad \text{(S20)}$$

$$\sum_{(n,i) \in Map_{nji}} A_{nji} + \sum_{(n,i) \in Map_{ijn}} A^{rev}_{nji} \leq \beta_j, \qquad \forall j \in J | j \notin RoI \quad \text{(S21)}$$

$$\sum_{(i,j) \in Map_{nji}} A_{nji} + \sum_{(i,j) \in Map_{ijn}} A^{rev}_{nji} \leq 0, \qquad \forall n | n = Eliminated\ Node \quad \text{(S22)}$$

Equations S17-S22 are identical to **S11-S15** with the following exceptions. First, the outer problem uses a different set of variables ($SINK, SOURCE, A$ and A^{rev}) than those used in the inner problem ($Sink, Source, a$ and a^{rev}). To emphasize this, all non-negative variables in the outer problem are capitalized. The RoI can not be deleted in the outer problem (**Eq. S20**), but is deleted in the inner problem using **Eq. S23,** along with other reactions the outer problem chooses to delete (by setting $\beta_j = 0$ in **Eq. 21**)

$$\beta_j = 0, \qquad \forall j \in RoI \quad \text{(S23)}$$

To ensure growth is still possible when the RoI is present, the following COBRA constraints (**Eq. S24-S29**) are included the outer problem:

$$\sum_{j \in J} S_{ij} \cdot v_j = 0, \qquad \forall i \in I \quad \text{(S24)}$$

$$v_j \leq v_j^{Upper} \cdot \beta_j \qquad \forall j \in J | j \notin RoI \quad \text{(S25)}$$

$$v_j \geq v_j^{Lower} \cdot \beta_j \qquad \forall j \in J | j \notin RoI \quad \text{(S26)}$$

$$v_j \leq v_j^{Upper} \qquad \forall j \in RoI \quad \text{(S27)}$$

$$v_j \geq v_j^{Lower} \qquad \forall j \in RoI \quad \text{(S28)}$$

$$v_{Biomass} \geq \mu^{min} \qquad \forall j \in RoI \quad \text{(S29)}$$

where I is the set of all metabolites, $v_{Biomass}$ is the biomass flux, and μ^{min} is the minimum biomass parameter, which was set to 0.1.

$$\max \sum_{n \in EndNode} SINK_n - 10 \cdot \sum_{n \in EndNode} Sink_n - 0.01 \cdot \sum_{j \in J} (1 - \beta_j) \quad \text{(S30)}$$

The inner (**Eq. S10-S15** with its dual) and outer (**Eq. S17-S29**) problems are combined and **Eq. S30** is used as an objective function to find a minimal set of reaction deletions (where each deletion is penalized by 0.01) such that a path exists in the outer problem where RoI is allowed, but no path exists in the inner problem where RoI is deleted along with other reactions in the minimal deletion set.

PathTracer sometimes finds incorrect paths (e.g., putrescine to glutamate but all carbon in glutamate comes from 2-oxoglutarate, see **Figure S1**) because individual carbon fates are not tracked (see Discussion). As a result, this bilevel approach might include additional un-necessary reaction deletions to eliminate incorrect paths. To eliminate the un-necessary reaction deletions, an MILP can be used to find the minimal essential set of PathTracer proposed reaction deletions, or the deleted reactions can be checked manually, by testing whether FBA still can not predict growth if the deleted reactions are added back.

4. PathTracer with Net Overall Transformations

PathTracer finds individual linear and cyclic paths; however, the algorithm can be modified to find all necessary paths needed so that the overall balanced reaction of the paths converts a starting metabolite (e.g., CO_2) into the ending metabolite (e.g., glutamate). Below we describe a formulation to find all paths, including internal loops, in one step. This method was applied to elucidate the CO_2 fixation pathway in iJO1366, which results in the net production of glutamate from CO_2, using putrescine as an energy and reductant source. The results were manually broken down into smaller sub-paths (**Tables S3-S11**) with their net overall reactions summarized in **Table S12** and **Figure S2**.

The modified algorithm is still a network flow problem. The node balances (**Eq. S31-S33**) are similar to those in the main text, except that $Sink$ and $Source$ variables are added to all nodes (defined as set I) not just the $StartNode$ and $EndNode$. This allows recycling of energy metabolites and completing reaction cycles that transfer non-carbon containing functional groups:

$$\sum_{(i,j) \in Map_{nji} | j \in For} a_{njid} + \sum_{(i,j) \in Map_{ijn} | j \in Rev} a^{rev}_{njid} = Source_n, \qquad \forall n \in I, d = 1 \quad \text{(S31)}$$

$$\sum_{(i,j) \in Map_{ijn} | j \in For} a_{ijnd} + \sum_{(i,j) \in Map_{nji} | j \in Rev} a^{rev}_{ijnd} = \sum_{(i,j) \in Map_{nji} | j \in For} a_{njid+1} + \sum_{(i,j) \in Map_{ijn} | j \in Rev} a^{rev}_{njid+1} Sink_{nd}, \quad \forall n \in I, d | 1 < d < |D| \quad \text{(S32)}$$

$$\sum_{(i,j) \in Map_{ijn} | j \in For} a_{ijnd} + \sum_{(i,j) \in Map_{nji} | j \in Rev} a^{rev}_{ijnd} = Sink_{nd}, \qquad \forall n \in I, d = |D| \quad \text{(S33)}$$

All sets and variables are defined as before except $Source_n$, $Sink_{nd}$, a and a^{rev} are now integer variables instead of binary variables. To reduce PathTracer's search space pFBA was used (**Eq. S1-S5)** to find reactions that are required to produce glutamate from carbon dioxide. The FBA solution was used to assign reactions to the sets For (if $v_j > 0$) or Rev (if $v_j < 0$) so that only edges associated with the active fluxes can be used. Additional constrains can be used to limit the number of times an edge can be used (**Eq. S34**) or to eliminate edges (**Eq. S35**).

$$\sum_{(n,i) \in Map_{nji} | j \in For} \sum_{d=1}^{|D|} a_{njid} + \sum_{(n,i) \in Map_{ijn} | j \in Rev} \sum_{d=1}^{|D|} a^{rev}_{njid} \leq L, \qquad \forall j \in J \quad \text{(S34)}$$

$$\sum_{(n,i) \in Map_{nji} | j \in For} \sum_{d=1}^{|D|} a_{njid} + \sum_{(n,i) \in Map_{ijn} | j \in Rev} \sum_{d=1}^{|D|} a^{rev}_{njid} \leq 0, \qquad \forall j | j = Eliminated\ Reaction \quad \text{(S35)}$$

Here L is the maximum of flow that can pass from node n to node i using reaction j. For the net production of glutamate from CO_2 and putrescine simulations, L was set to fifty and the eliminated

reactions included SSALx, SSALy, SPMS, GGPTRCS, CBMKr, HCO3E and POR5 (see main text for details).

The modified PathTracer algorithm finds a minimal set of carbon movements that result in the net production of products ($EndMetabs$) from a defined set of reactants ($StartMetabs$). To enforce an overall net transformation the PathTracer algorithm use the following constraints (**Eq. S36-S39**):

$$S_i^{Net} = \sum_{(n,j,i)\in Map_{nji} | j\in For} \sum_{d=1}^{|D|} S_{ij}.a_{njid} + \sum_{(n,j,i)\in Map_{ijn} | j\in Rev} \sum_{d=1}^{|D|} -S_{ij}.a_{njid}^{rev}, \qquad \forall i \in I \tag{S36}$$

$$S_i^{Net} \leq -1, \quad \forall i \in StartMetabs \tag{S37}$$

$$S_i^{Net} \geq -1, \quad \forall i \in EndMetabs \tag{S38}$$

$$S_i^{Net} = 0, \quad \forall i \in I | i \notin (StartMetabs \cup EndMetabs \cup SmallMetabs) \tag{S39}$$

where S_i^{Net} is a free variable that represents the stoichiometric coefficient of each metabolite, i, in the net reaction and is calculated using **Eq. S36**. The remaining constraints (**Eq. S37**-**S39**) ensure that the net reaction consumes $StartMetabs$ and produces $EndMetabs$ without producing or consuming any other metabolites, except those included in $SmallMetabs$. $StartMetabs$ contains all carbon containing starting metabolites, $EndMetabs$ contains all carbon containing end-products, and $SmallMetabs$ contains all non-carbon metabolites that can balance the net reaction. All three sets can be defined using FBA simulation results. For the CO_2 to glutamate case, $StartMetabs$ contained extracellular putrescine and extracellular CO_2, $EndMetabs$ contained extracellular 4-amino-butanoate and cytoplasmic glutamate, and $SmallMetabs$ included extracellular, cytoplasmic, and periplasmic forms of NH_4, H_2O, H, and O_2, as well as cytoplasmic forms of PO_4, and diphosphate.

To find solutions using sinks and sources associated with just the starting and ending metabolites, penalties were applied to sources and sinks for other metabolites. In this case, penalties punish using a new sink or source but not the amount of flow going through a given source or sink (since this tended to produce long cyclic paths). Penalties were enforced using constraints (**Eq. S40-S41**), where $SourcePenalty$ and $SinkPenalty$ are binary variables indicating if a source or sink is being used.

$$Source_n \leq 100 \cdot SourcePenalty_n, \qquad \forall n \in I | n \notin StartMetabs \tag{S40}$$

$$Sink_{nd} \leq 100 \cdot SinkPenalty_{nd}, \qquad \forall (n,d) \in (I,D) \tag{S41}$$

Using constraints **S31-S41**, PathTracer finds the shortest set of carbon transfer paths which generate an appropriate S^{Net} with the fewest source and sink penalties by minimizing the following objective:

$$\max \sum_{n\in I} \sum_{d=1}^{|D|} d \cdot \theta_n \cdot SinkPenalty_{nd} - 100 \cdot \sum_{n\in I} SourcePenalty_n \tag{S42}$$

Here θ_n is the penalty for using a particular sink, where θ_n = -0.01 if $n \in EndMetabs$ and θ_n = -1 if $n \notin EndMetabs$. The MILP was formulated using the general algebraic modeling system (GAMS) and solved using CPLEX 12. The algorithm was allowed to run for twenty minutes or until optimality, whichever came first. For this problem, feasibility is more important than optimality as any solution meets the basic criteria of converting CO_2 to glutamate. In this instance, optimality just improves the quality of the path output. One solution that was found is shown in **Figure S2** and summarized in **Tables S3-S12**.

Supplementary Figures

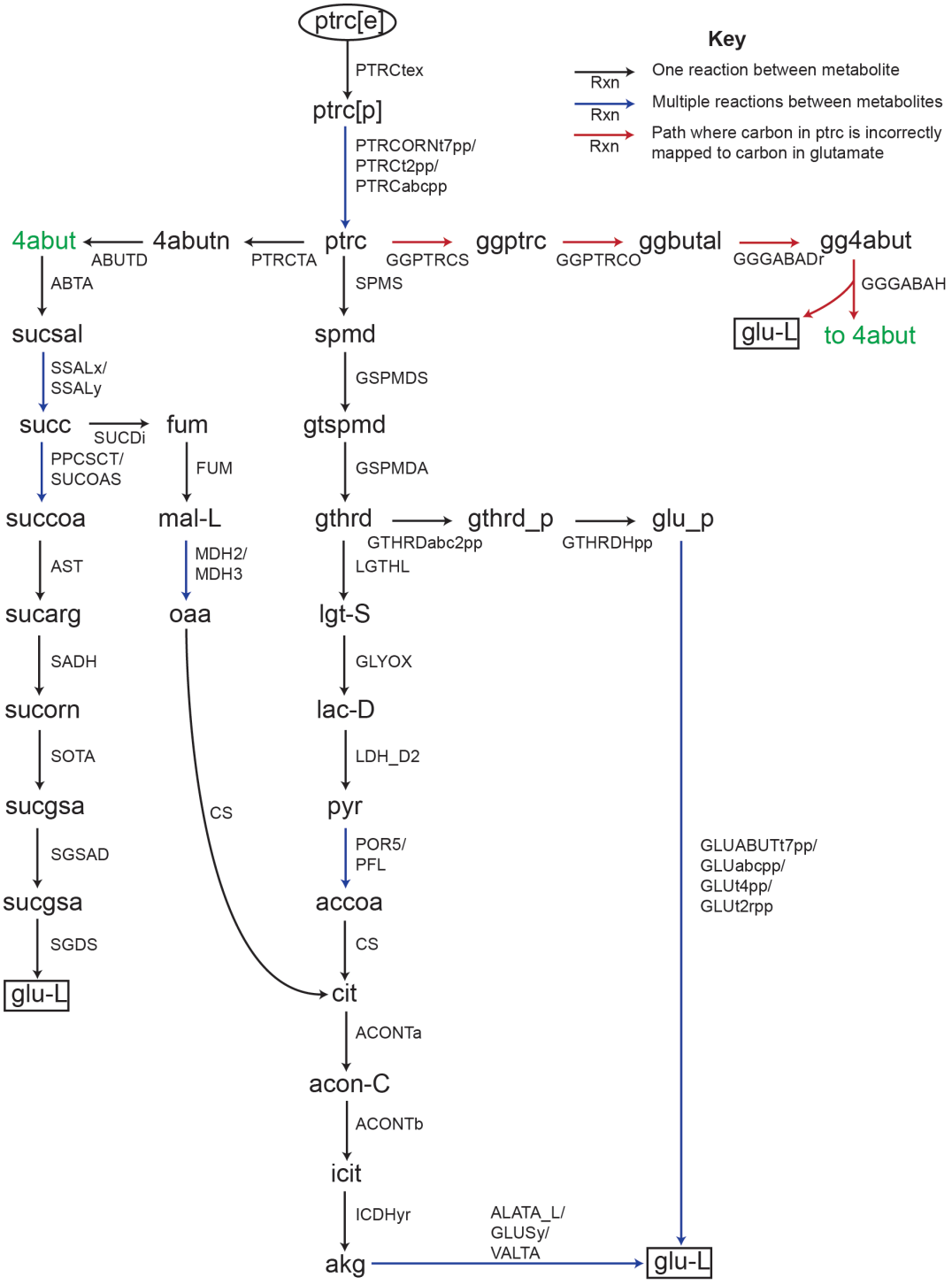

Figure S1. Top 15 Shortest Paths from Putrescine to Glutamate.
PathTracer was used to find paths from putrescine (ptrc) to L-glutamate (glu-L). The top fifteen shortest are shown above. Blue lines indicate that multiple reactions can connect a pair of metabolites. Circled and squared metabolites indicate the starting and ending nodes for PathTracer, respectively. 4abut is highlighted in green to indicate it is shown twice. Metabolite abbreviations match those from iJO1366.

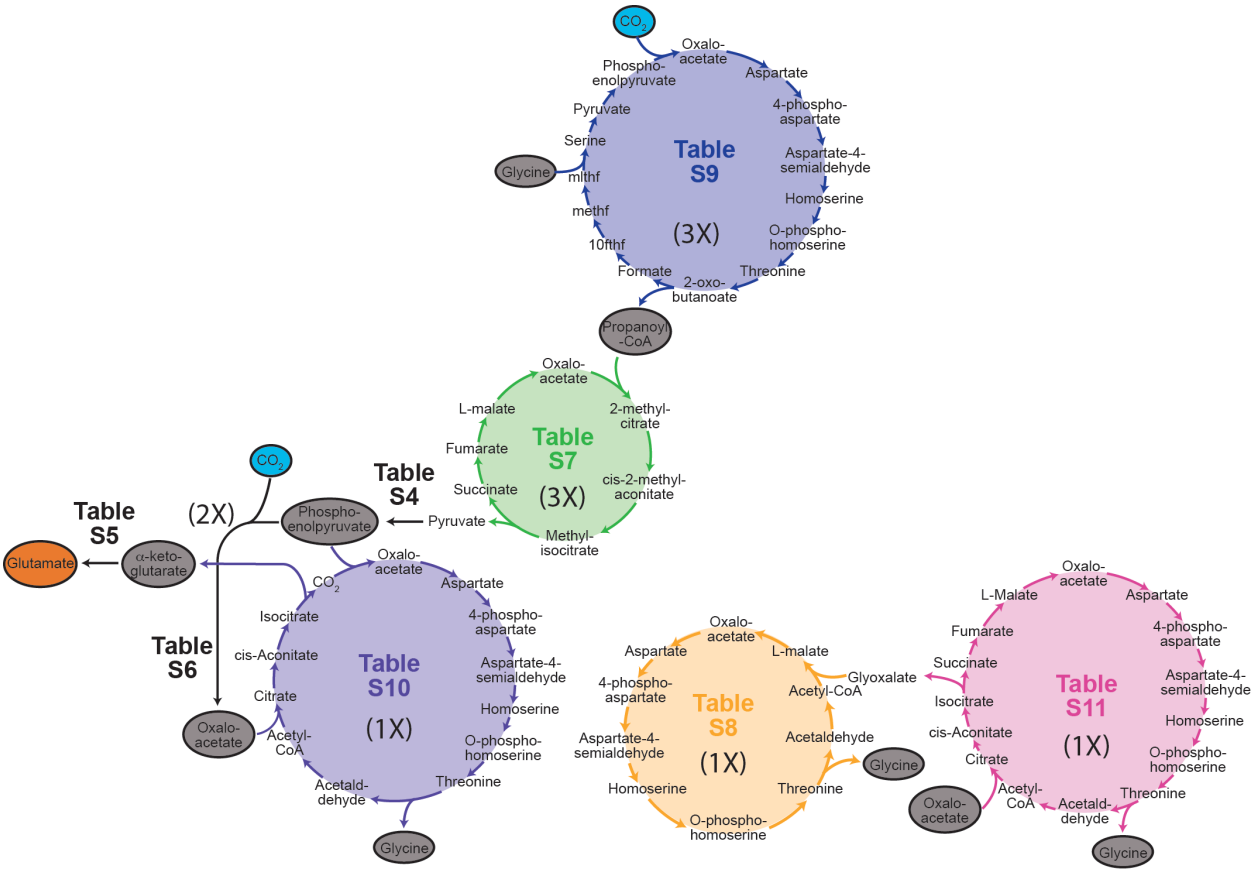

Figure S2. Carbon Sub-Paths Involved in the Net Transformation of CO_2 to Glutamate.
PathTracer was used to find paths involved in the overall net transformation of CO2 to glutatmate, using putrescine as electron and energy source. The identified path was broken down into different sub-paths that are summarized in **Tables S3-S11**. All sub-paths are shown except for the respiratory sub-path (**Table S3**) that recycles electron carriers generating a proton gradient. For simplicity not all reactants and products are shown, but the overall reaction for each sub-path can be found in **Table S12**. The cycle involved in **Table S9** is the same as that used in the CO_2 to acetyl-CoA path (labeled as OAA Cycle 1 in **Figure 5**).

Supplementary Tables

*All metabolite and reaction abbreviations used in supplementary tables (S2-S12) match those used in iJO1366.

Table S1. iJO1366 Carbon Transfer Map Category and MapMaker Performance Statistics.

Reaction Type	**No. (%) of Reactions**	**No. (%) of Incorrect Carbon Maps**
Isomerization/Transport/Non-Carbon Modifications		
w/o Energy Carriers	921 (40.9%)	0 (0%)
w/ Energy Carriers	467 (20.7%)	0 (0%)
Combination		
w/o Energy Carriers	70 (3.1%)	0 (0%)
w/ Energy Carriers	67 (3.0%)	0 (0%)
Decomposition / (A+A)		
w/o Energy Carriers	337 (15.0%)	0 (0%)
w/ Energy Carriers	12 (0.5%)	0 (0%)
Facilitated Decomposition		
w/o Energy Carriers	4 (0.2%)	0 (0%)
w/ Energy Carriers	1 (0.04%)	0 (0%)
Substitution (Non-Carbon Containing)/Cotransporters		
w/o Energy Carriers	88 (3.9%)	12 (14%)
w/ Energy Carriers	6 (0.3%)	0 (0%)
Substitution (Carbon Containing)		
w/o Energy Carriers	161 (7.1%)	43 (27%)
w/ Energy Carriers	3 (0.1%)	0 (0%)
Higher Complexity	21 (0.9%)	11 (52%)
Inorganic Reactions	93 (4.1%)	0 (0%)
Total	2251 (100%)	66 (2.9%)

Table S2. iJO1366 Reactions with Incorrectly Predicted Carbon Transfer Maps

3OAS100	h[c] + malACP[c] + ocACP[c] -> 3odecACP[c] + ACP[c] + co2[c]
3OAS120	dcaACP[c] + h[c] + malACP[c] -> 3oddecACP[c] + ACP[c] + co2[c]
3OAS121	cdec3eACP[c] + h[c] + malACP[c] -> 3ocddec5eACP[c] + ACP[c] + co2[c]
3OAS140	ddcaACP[c] + h[c] + malACP[c] -> 3omrsACP[c] + ACP[c] + co2[c]
3OAS141	cddec5eACP[c] + h[c] + malACP[c] -> 3ocmrs7eACP[c] + ACP[c] + co2[c]
3OAS181	h[c] + hdeACP[c] + malACP[c] -> 3ocvac11eACP[c] + ACP[c] + co2[c]
3OAS60	butACP[c] + h[c] + malACP[c] -> 3ohexACP[c] + ACP[c] + co2[c]
3OAS80	h[c] + hexACP[c] + malACP[c] -> 3ooctACP[c] + ACP[c] + co2[c]
ACACCT	acac[c] + accoa[c] -> aacoa[c] + ac[c]
ACACT2r	accoa[c] + btcoa[c] <=> 3ohcoa[c] + coa[c]
ACACT3r	accoa[c] + hxcoa[c] <=> 3oocoa[c] + coa[c]
ACACT4r	accoa[c] + occoa[c] <=> 3odcoa[c] + coa[c]
ACACT5r	accoa[c] + dcacoa[c] <=> 3oddcoa[c] + coa[c]
ACACT6r	accoa[c] + ddcacoa[c] <=> 3otdcoa[c] + coa[c]
ACACT7r	accoa[c] + tdcoa[c] <=> 3ohdcoa[c] + coa[c]
ADNK1	adn[c] + atp[c] -> adp[c] + amp[c] + h[c]
ADOCBLS	agdpcbi[c] + rdmbzi[c] -> adocbl[c] + gmp[c] + h[c]
ADSK	aps[c] + atp[c] -> adp[c] + h[c] + paps[c]
BMOCOS	moco[c] + mptamp[c] -> amp[c] + bmoco[c] + cu2[c]
BUTCT	accoa[c] + but[c] -> ac[c] + btcoa[c]
BWCOS	mptamp[c] + wco[c] -> amp[c] + bwco[c] + cu2[c]
CRNBTCT	bbtcoa[c] + crn[c] <=> crncoa[c] + gbbtn[c]
CRNCBCT	crn[c] + ctbtcoa[c] <=> crncoa[c] + ctbt[c]
CRNt8pp	crn-D[c] + crn[p] -> crn[c] + crn-D[p]
DGK1	atp[c] + dgmp[c] <=> adp[c] + dgdp[c]
DHAPT	dha[c] + pep[c] -> dhap[c] + pyr[c]
DXPS	g3p[c] + h[c] + pyr[c] -> co2[c] + dxyl5p[c]
ECAP2pp	eca2und[p] + unagamuf[p] -> eca3und[p] + h[p] + udcpdp[p]
ECAP3pp	eca3und[p] + unagamuf[p] -> eca4und[p] + h[p] + udcpdp[p]
ENTCS	3 23dhba[c] + 3 seramp[c] -> 6 amp[c] + enter[c] + 9 h[c]
FORCT	forcoa[c] + oxa[c] <=> for[c] + oxalcoa[c]
GDPDPK	atp[c] + gdp[c] -> amp[c] + h[c] + ppgpp[c]
GK1	atp[c] + gmp[c] <=> adp[c] + gdp[c]
HXCT	accoa[c] + hxa[c] -> ac[c] + hxcoa[c]
MPTG2	uaagmda[c] + murein5p5p[p] -> h[c] + udcpdp[c] + murein5p5p5p[p]
NDPK1	atp[c] + gdp[c] <=> adp[c] + gtp[c]
NDPK5	atp[c] + dgdp[c] <=> adp[c] + dgtp[c]
NDPK8	atp[c] + dadp[c] <=> adp[c] + datp[c]
NNDMBRT	dmbzid[c] + nicrnt[c] -> 5prdmbz[c] + h[c] + nac[c]
O16AP2pp	o16a2und[p] + o16aund[p] -> h[p] + o16a3und[p] + udcpdp[p]
O16AP3pp	o16a3und[p] + o16aund[p] -> h[p] + o16a4und[p] + udcpdp[p]
OPMEACPS	gmeACP[c] + h[c] + malACP[c] -> ACP[c] + co2[c] + opmeACP[c]
PAPPT3	udcpp[c] + ugmda[c] -> uagmda[c] + ump[c]
PDH	coa[c] + nad[c] + pyr[c] -> accoa[c] + co2[c] + nadh[c]
PGSA120	cdpdddecg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp120[c]
PGSA140	cdpdtdecg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp140[c]
PGSA141	cdpdtdec7eg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp141[c]
PGSA160	cdpdhdecg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp160[c]
PGSA161	cdpdhdec9eg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp161[c]
PGSA180	cdpdodecg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp180[c]

PGSA181	cdpdodec11eg[c] + glyc3p[c] -> cmp[c] + h[c] + pgp181[c]
PPCSCT	ppcoa[c] + succ[c] -> ppa[c] + succoa[c]
PSSA120	cdpdddecg[c] + ser-L[c] -> cmp[c] + h[c] + ps120[c]
PSSA140	cdpdtdecg[c] + ser-L[c] -> cmp[c] + h[c] + ps140[c]
PSSA141	cdpdtdec7eg[c] + ser-L[c] -> cmp[c] + h[c] + ps141[c]
PSSA160	cdpdhdecg[c] + ser-L[c] -> cmp[c] + h[c] + ps160[c]
PSSA161	cdpdhdec9eg[c] + ser-L[c] -> cmp[c] + h[c] + ps161[c]
PSSA180	cdpdodecg[c] + ser-L[c] -> cmp[c] + h[c] + ps180[c]
PSSA181	cdpdodec11eg[c] + ser-L[c] -> cmp[c] + h[c] + ps181[c]
SEPHCHCS	akg[c] + h[c] + ichor[c] -> 2sephchc[c] + co2[c]
SHSL1	cys-L[c] + suchms[c] -> cyst-L[c] + h[c] + succ[c]
SUCFUMtpp	succ[c] + fum[p] <=> fum[c] + succ[p]
TALA	g3p[c] + s7p[c] <=> e4p[c] + f6p[c]
TRE6PS	g6p[c] + udpg[c] -> h[c] + tre6p[c] + udp[c]
UGLT	gal1p[c] + udpg[c] <=> g1p[c] + udpgal[c]
VALTA	akg[c] + val-L[c] <=> 3mob[c] + glu-L[c]

Table S3. NADH and Ubiquinone Recycling

Depth	Path	Flow
1	q8h2 --> q8 via CYTBO3_4pp	33
2	q8 --> q8h2 via NADH16pp	29

Table S4. ATP Generation and Pyruvate to Phosphoenolpyruvate

Depth	Path	Flow
1	adp --> atp via ATPS4rpp	27
2	atp --> adp via ADK1	6
2	atp --> amp via PPS	3

Table S5. Glutamate Generation and Putrescine Degradation

Depth	Path	Flow
1	ptrc_e --> ptrc_p via PTRCtex	21
2	ptrc_p --> ptrc via PTRCt2pp	21
3	ptrc --> 4abutn via PTRCTA	21
4	4abutn --> 4abut via ABUTD	21
5	4abut --> 4abut_p via GLUABUTt7pp	21
6	4abut_p --> 4abut_e via ABUTtex	21
1	glu-L --> akg via GLUDy	14
1	glu-L --> glu-L_p via GLUt2rpp	21

Table S6. CO_2 and Phosphoenolpyruvate to Oxaloacetate

Depth	Path	Flow
1	co2_e --> co2_p via CO2tex	5
2	co2_p --> co2 via CO2tpp	5
3	co2 --> oaa via PPC	2

Table S7. Propanoyl-CoA to Pyruvate

Depth	Path	Flow
4,36	oaa --> 2mcit via MCITS	3
5,37	2mcit --> 2mcacn via MCITD	3
6,38	2mcacn --> micit via MICITDr	3
7,39	micit --> succ via MCITL2	3
8,40	succ --> fum via SUCDi	3
9,41	fum --> mal-L via FUM	3
10,42	mal-L --> oaa via MDH	3

Table S8. Glyoxylate to Glycine

Depth	Path	Flow
11	oaa --> asp-L via ASPTA	1
12	asp-L --> 4pasp via ASPK	1
13	4pasp --> aspsa via ASAD	1
14	aspsa --> hom-L via HSDy	1
15	hom-L --> phom via HSK	1
16	phom --> thr-L via THRS	1
17	thr-L --> acald via THRAi	1
18	acald --> accoa via ACALD	1
19	accoa --> mal-L via MALS	1
20	mal-L --> oaa via MDH	1

Table S9. CO_2 and Glycine to Propanoyl-CoA

Depth	Path	Flow
11,21,41	oaa --> asp-L via ASPTA	3
12,22,42	asp-L --> 4pasp via ASPK	3
13,23,43	4pasp --> aspsa via ASAD	3
14,24,44	aspsa --> hom-L via HSDy	3
15,25,45	hom-L --> phom via HSK	3
16,26,46	phom --> thr-L via THRS	3
17,27,47	thr-L --> 2obut via THRD_L	3
18,28,48	2obut --> formate via OBTFL	3
19,29,49	formate --> 10fthf via FTHFLi	3
20,30,50	10fthf --> methf via MTHFC	3
21,31,51	methf --> mlthf via MTHFD	3
22,32,52	mlthf --> ser-L via GHMT2r	3
23,33,53	ser-L --> pyr via SERD_L	3
24,34,54	pyr --> pep via PPS	3
25,35,55	pep --> oaa via PPC	3

Table S10. Oxaloacetate and Phosphoenolpyruvate to α-Ketoglutarate and Glycine

Depth	Path	Flow
43	oaa --> asp-L via ASPTA	1
44	asp-L --> 4pasp via ASPK	1
45	4pasp --> aspsa via ASAD	1
46	aspsa --> hom-L via HSDy	1
47	hom-L --> phom via HSK	1
48	phom --> thr-L via THRS	1
49	thr-L --> acald via THRAi	1
50	acald --> accoa via ACALD	1
51	accoa --> cit via CS	1
52	cit --> acon-C via ACONTa	1
53	acon-C --> icit via ACONTb	1
54	icit --> co2 via ICDHyr	1
55	co2 --> oaa via PPC	1

Table S11. Oxaloacetate to Glycine and Glyoxylate

Depth	Path	Flow
26	oaa --> asp-L via ASPTA	1
27	asp-L --> 4pasp via ASPK	1
28	4pasp --> aspsa via ASAD	1
29	aspsa --> hom-L via HSDy	1
30	hom-L --> phom via HSK	1
31	phom --> thr-L via THRS	1
32	thr-L --> acald via THRAi	1
33	acald --> accoa via ACALD	1
34	accoa --> cit via CS	1
35	cit --> acon-C via ACONTa	1
36	acon-C --> icit via ACONTb	1
37	icit --> succ via ICL	1
38	succ --> fum via SUCDi	1
39	fum --> mal-L via FUM	1
40	mal-L --> oaa via MDH	1

Table S12. Overall Reactions for Each Sub-Path and Net Overall Transformation Reaction

Sub-Path Table	Overall Sub-Path or Net Reactant Stoichiometry	Overall Sub-Path or Net Product Stoichiometry
S3	248 h + 29 nadh + 16.5 o2 + 4 q8h2	219 h[p] + 33 h2o + 29 nad + 4 q8
S4	15 adp + 3 amp + 108 h[p] + 24 pi + 3 pyr	18 atp + 87 h + 24 h2o + 3 pep
S5	7 akg + 35 h2o + 21 nad + 14 nadp + 21 ptrc[e]	21 4abut[e] + 7 glu-L + 56 h + 21 nadh + 14 nadph + 14 nh4
S6	5 co2[e] + 2 h2o + 2 pep	3 co2 + 2 h + 2 oaa + 2 pi
S7	6 h2o + 3 nad + 3 ppcoa + 3 q8	3 coa + 6 h + 3 nadh + 3 pyr + 3 q8h2
S8	2 atp + 1 glu-L + 1 glx + 2 h2o + 2 nad + 2 nadph	2 adp + 1 akg + 1 gly + 2 h + 2 nadh + 2 nadp + 2 pi
S9	12 atp + 3 co2 + 3 coa + 3 gluL + 3 gly + 9 h2o + 9 nadph	9 adp + 3 akg + 3 amp + 3 h + 9 nadp + 6 nh4 + 15 pi + 3 ppcoa
S10	2 atp + 1 glu-L + 3 h2o + 1 nad + 1 nadph + 1 oaa + 1 pep	2 adp + 2 akg + 1 gly + 2 h + 1 nadh + 1 nadp + 3 pi
S11	2 atp + 1 glu-L + 3 h2o + 2 nad + 2 nadph + 1 oaa + 1 q8	2 adp + 1 akg + 1 glx + 1 gly + 2 h + 2 nadh + 2 nadp + 2 pi + 1 q8h2
Net Overall Rxn	5 co2[e] + 88 h + 3 h2o + 16.5 o2 + 21 ptrc[e]	21 4abut[e] + 1 glu-L + 111 h[p] + 20 nh4