**Supporting Text S1**

## 0th degree concentration change coupling

0degree.tif

The Michaelis-Menten kinetics equation is given by: 

Rearranging for S gives:

 (1)

For the reference condition: 

Dividing Eq.1 by the reference condition equation gives:

 (2)

Assuming that V<<Vmax & V\*<<Vmax\* (see main text for the discussion on the validity of the assumption) one gets:

 (3)

Transforming to log-space Eq. 3 becomes:



since Vmax = k2[E], where k2 and [E] are, respectively, substrate to product conversion rate and concentration of active enzyme. Assuming that E ∝ T, where T is transcript abundance, and assuming that k2 does not change between the two conditions, one gets:



or

 (4)

Eq. 4 is defined as 0th degree concentration change coupling.

## 1st degree concentration change coupling§

1degree.tif

At steady state Vin =Vout.Consequently,



Substituting the flux ratio in the above equation with metabolite and transcript ratios, as given by Eq.4, we obtain:

,

Since the consuming reaction of R is the same as the production reaction of S, .

Rearranging for S therefore gives:

 (5)

Eq. 5 is defined as 1st degree concentration change coupling.

§In the presented scheme, Umax is equivalent to

## 2nd degree concentration change coupling§

## 2degree.tif

Applying the first degree CoCCoA (Eq.5) to metabolite R, gives:

 (6)

Since, substituting the *R* term from Eq.5 into Eq.6 gives:

 (7)

Eq. 7 is defined as 2nd degree concentration change coupling.

§In the presented scheme, *Umax* is equivalent to and *Wmax* is the capacity constraint for the production of metabolite R

## Multiple reactions connected to S

### 0th degree concentration change coupling

2reactions.tif

Two reactions using the same substrate *S*. *U* and *V* denote fluxes through the two reactions.

Consider two reactions (carrying flux *U* and *V*) that use the same substrate *S*. For each reaction, 0th degree concentration change coupling (Eq.4) can be applied independently



The above system of two equations can be summed and rearranged as:

 (8)

For more than two reactions, similar analysis will imply averaging of fold changes of the corresponding transcripts.

### 1st degree concentration change coupling

3reactions.tif

Multiple reactions producing the same metabolite *S*.

At steady state:



Comparing to the reference condition:





Define *f1*=*αf2,* *β = 1/α*, *f1\**=*α\*f2\** and *β\*= 1/α\**:



In cases where *α=α\** and *β=β\**, meaning that the split ratio of fluxes between conditions is unchanging (for example, as suggested in [[1](#_ENREF_1)]), we obtain:



 (9)

Rearrangement of Eq.9 using Eq.4 gives:

 (10)

Equation 10 was used as a basis for calculation of the 1st degree concentration change couplings.

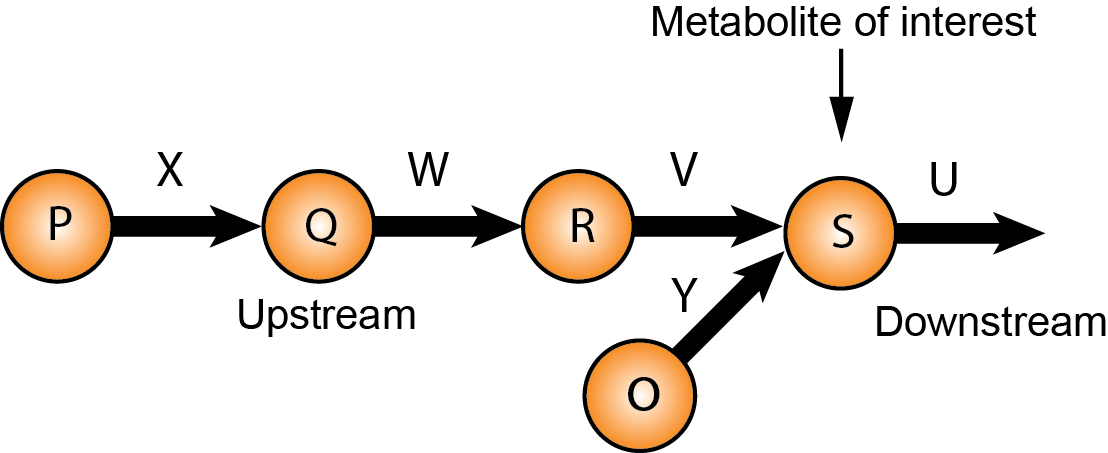
### 1st degree concentration change coupling with protein-mRNA correlation correction factor



Each transcript change term  was multiplied by a correction factor *β*,which was randomly sampled from a normal distribution with mean and variance estimated based on the values of the slopes of the least squares regression lines for the mRNA-protein fold change data (Supplementary Figure S2).

## Alternative CoCCoA formulation

An alternative formulation of the higher-degree CoCCoA equations includes information from all the intermediate reaction steps till the desired degree. This formulation takes in to account mass balance around metabolites within the desired distance from the metabolite of interest. An example for this formulation is illustrated below where three upstream and one downstream degrees are considered.





The above system in log space can be rewritten as:



Combining the above equation with Eq. 4 and applying the same assumptions as in Eq. 9 gives:

 (9)

The above-described method can be applied in the same fashion to derive the equations including the pathways downstream from the metabolite of interest.

### Algorithm for calculation of CoCCoA scores in the alternative formulation

The procedure below describes the algorithm used for computing the scores for the alternative CoCCoA formulation. We note that this algorithm is a heuristic and does not rigorously check for consistency with the mass balances. However, this is of minor concern since several reactions need to be removed from the metabolic network due to the uncertainty in their flux directions, and consequently the final network is not necessarily flux balanced.

INPUT:

CoCCoA\_degree ∈ N, where N = {|a| > 0: a ∈ Z}

Bipartite directed graph G = (U,V,E) of metabolic network where U is a set of metabolite nodes, V is a set of reactions nodes and E set of edges between them.

Direction of scoring D = {upstream or downstream}

RFC – dictionary where keys are reactions and values are fold-change of reaction

MFC - dictionary where keys are metabolites and values are fold-change of metabolite (optional)

RFC.keys ∈ V, MFC.keys ∈ U

OUTPUT:

scores for all metabolites at given distance

GU = get\_unipartite\_graph(G, nodetype=metabolites) //returns a directed unipartite projection (metabolite graph)

IF *direction* is upstream

GU = reverse(GU) #reverses edge direction

FOR each *u* in U,

T = empty dictionary with key-value pairs

FOR each *d* in 1:CoCCoA\_degree

*targets* = find\_target\_nodes(in= GU, from=*u*, at\_distance=*d*), *targets* ∈ U

T = (key = *d*, value = *targets*)

PATHS = empty dictionary with key-value pairs

FOR each key-value pair (*d*, *targets*) in T

a)P = find\_all\_simple\_reaction\_paths(in=G, from=*u*, to=*targets*, of\_length=*d*+1)),

P ≡ //returns: pathways of reactions at distance d

PATHS = (key = *d*, value = P)

FOR each *d* in 1:CoCCOA\_degree:

score = 0, metabolite score at distance *d*

PATHS\_f = empty dictionary with key-value pairs

PATHS\_f = remove\_subpathways(from=PATHS, until\_distance=*d*), , where *i* is distance and PATHS\_f ∈ PATHS //if path at *i* distance is a subset of any path at *i*+1 (including *i* itself),then it is removed; pathways at distance more than *d* are removed; returns: key – distance ≤ *d*, value remained unique pathways at distance

R\_counts = empty dictionary with key-value pairs

R\_counts = count\_reactions\_in\_paths(PATHS\_f)//returns: key - reaction, value - number of times it was present in PATHS\_f

R\_path\_lenghts = empty dictionary with key-value pairs

R\_path\_lenghts = count\_reactions\_in\_paths(PATHS\_f)//returns: key – reaction, value – length of

shortest path where reaction was found in PATHS\_f

R = empty array

R = get\_reactions(from=PATHS\_f, until\_distance=d), R ∈ V //returns: unique reactions from PATH\_f

FOR each *r* in R, R ∈ V

r\_weight = R\_counts[*r*]/total(PATHS\_f.values())/R\_path\_lenghts[*r*]

IF *r* is in RFC

score += RFC[*r*]\*r\_weight

IF MFC is not NULL //adds metabolic component

m\_neighbors = empty array

IF *direction* is upstream

m\_neighbors = get\_output\_nodes(in=G, source=*r*), m\_neighbors ∈ U

ELSE m\_neighbors = get\_input\_nodes(in=G, source=*r*) , m\_neighbors ∈ U

FOR each *m* in m\_neighbors, *m* ≠ *u*

IF *m* is in MFC

score += MFC[m]\*r\_weight/length(m\_neighbours)

ELSE score += mean(MFC)\*r\_weight/length(m\_neighbours)

b)//subtracts connected fluxes which are not part of PATHS\_f

IF *r* was never found last in elements of PATHS\_f

m\_neighbors = empty array

IF *direction* is upstream

m\_neighbors = get\_output\_nodes(in=G, source=*r*), m\_neighbors ∈ U

ELSE m\_neighbors = get\_input\_nodes(in=G, source=*r*), m\_neighbors ∈ U

FOR each *m* in m\_neighbours, *m* ≠ *u*, *m* is never a neighbor of elements in R

r\_neighbors = empty array

IF *direction* is upstream:

r\_neighbors = get\_input\_nodes(of=*m*, from=G), r\_neigbours ∈ V & r\_neigbours ∉ R

ELSE r\_neighbors = get\_output\_nodes(of=*m*, from=G), r\_neigbours ∈ V & r\_neigbours ∉ R

(continued 5 indentations)

IF MFC is not NULL

IF *m* is in MFC

score -= MFC[m]\*r\_weight/length(m\_neighbours)

ELSE

score -= mean(MFC)\*r\_weight/length(m\_neighbours)

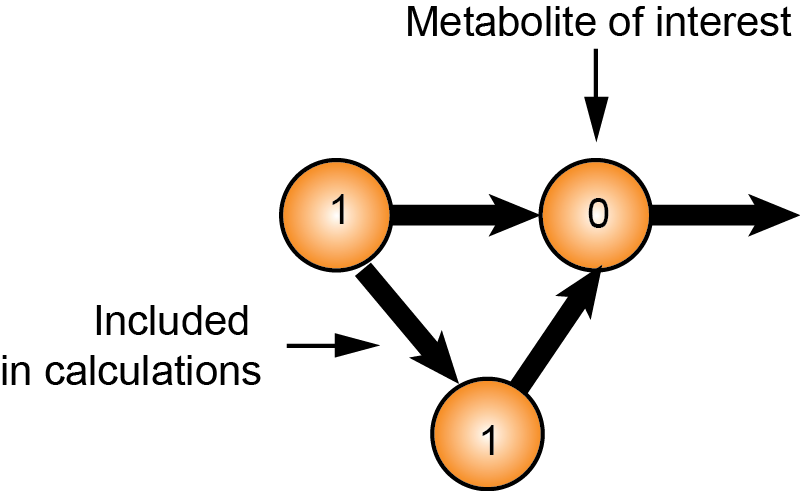
FOR each *n* in r\_neighbours

score -= RFC[r]\*r\_weight

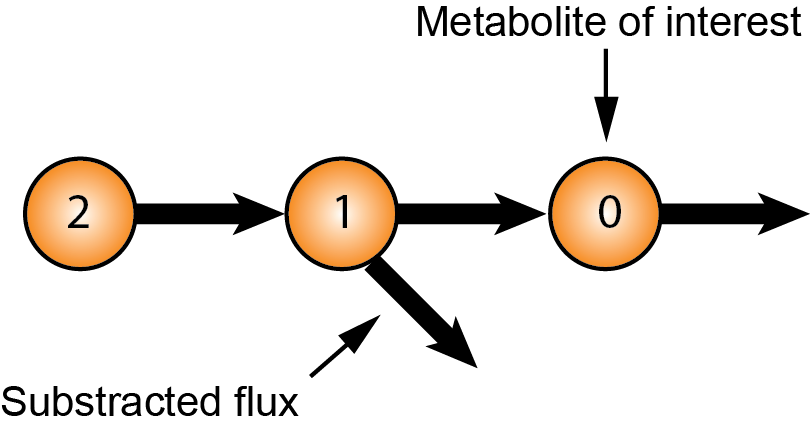
save score

The above procedure weights fluxes based on the frequency of their appearance in the paths starting from the metabolite of interest and leading to the target nodes at the desired distance. Additionally, all pathway scores are normalized by their lengths. Some of the key considerations from the algorithm implementation are listed below:

1. The procedure accounts for all paths starting from the metabolite of interest and up to the ‘desired distance’ + 1. The extra distance is tracked so as to account for the triangles in the network, e.g. as shown below:



1. The procedure subtracts the contribution of fluxes that are not part of the paths, but are part of the intermediate metabolite mass balances. The ‘last’ metabolites in the pathway are not considered.



## Error function for Figure S1.

 (11)

# References (Supporting information)

1. Haverkorn van Rijsewijk BR, Nanchen A, Nallet S, Kleijn RJ, Sauer U (2011) Large-scale 13C-flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in Escherichia coli. Mol Syst Biol 7: 477.

2. Forster J, Famili I, Fu P, Palsson BO, Nielsen J (2003) Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. Genome Res 13: 244-253.

3. Usaite R, Nielsen J, Olsson L (2008) Physiological characterization of glucose repression in the strains with SNF1 and SNF4 genes deleted. J Biotechnol 133: 73-81.

4. Usaite R, Wohlschlegel J, Venable JD, Park SK, Nielsen J, et al. (2008) Characterization of global yeast quantitative proteome data generated from the wild-type and glucose repression saccharomyces cerevisiae strains: the comparison of two quantitative methods. J Proteome Res 7: 266-275.

5. Griffin TJ, Gygi SP, Ideker T, Rist B, Eng J, et al. (2002) Complementary profiling of gene expression at the transcriptome and proteome levels in Saccharomyces cerevisiae. Mol Cell Proteomics 1: 323-333.

6. Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, et al. (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. Science 292: 929-934.

7. Fendt SM, Buescher JM, Rudroff F, Picotti P, Zamboni N, et al. (2010) Tradeoff between enzyme and metabolite efficiency maintains metabolic homeostasis upon perturbations in enzyme capacity. Mol Syst Biol 6: 356.

8. Kresnowati MT, van Winden WA, Almering MJ, ten Pierick A, Ras C, et al. (2006) When transcriptome meets metabolome: fast cellular responses of yeast to sudden relief of glucose limitation. Mol Syst Biol 2: 49.

9. Wisselink HW, Cipollina C, Oud B, Crimi B, Heijnen JJ, et al. (2010) Metabolome, transcriptome and metabolic flux analysis of arabinose fermentation by engineered Saccharomyces cerevisiae. Metab Eng 12: 537-551.

10. Tu BP, Kudlicki A, Rowicka M, McKnight SL (2005) Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. Science 310: 1152-1158.

11. Tu BP, Mohler RE, Liu JC, Dombek KM, Young ET, et al. (2007) Cyclic changes in metabolic state during the life of a yeast cell. Proc Natl Acad Sci U S A 104: 16886-16891.