Goldstein and Bockmayr *Algorithms for Molecular Biology* (2015) 10:1
DOI 10.1186/s13015-014-0028-y

ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

Double and multiple knockout simulations for genome-scale metabolic network reconstructions

Yaron AB Goldstein* and Alexander Bockmayr

Abstract

Background: Constraint-based modeling of genome-scale metabolic network reconstructions has become a widely used approach in computational biology. Flux coupling analysis is a constraint-based method that analyses the impact of single reaction knockouts on other reactions in the network.

Results: We present an extension of flux coupling analysis for double and multiple gene or reaction knockouts, and develop corresponding algorithms for an *in silico* simulation. To evaluate our method, we perform a full single and double knockout analysis on a selection of genome-scale metabolic network reconstructions and compare the results.

Software: A prototype implementation of double knockout simulation is available at http://hoverboard.io/L4FC.

Keywords: Constraint-based modeling, Metabolic network, Flux coupling analysis, Reaction knockout, Gene knockout

Background

Constraint-based modeling has become a widely used approach for the analysis of genome-scale reconstructions of metabolic networks [1]. Given a set of metabolites $\mathcal{M}$ and a set of reactions $\mathcal{R}$, the metabolic network is represented by its stoichiometric matrix $S \in \mathbb{R}^{\mathcal{M}\times\mathcal{R}}$, and a subset of irreversible reactions $\mathrm{Irr} \subseteq \mathcal{R}$. The flux cone $C = \{v \in \mathbb{R}^{\mathcal{R}} \mid Sv = 0, v_r \geq 0, r \in \mathrm{Irr}\}$ contains all steady-state flux vectors satisfying the stoichiometric and thermodynamic irreversibility constraints. Based on this flux cone, many analysis methods have been proposed over the years (see e.g. [2] for an overview). *Flux Balance Analysis* (FBA) [3,4] solves a linear program (LP) $\max\{z(v) \mid Sv = 0, l \leq v \leq u\}$ over the (truncated) flux cone in order to predict how efficiently an organism can realize a certain biological objective, represented by the linear objective function $z(v)$. For example, one may compute the maximal biomass production rate under some specific growth conditions. *Flux Coupling Analysis* (FCA) [5,6] studies dependencies between reactions. Here the question is whether or not for all steady-state flux vectors $v \in C$, zero flux $v_r = 0$ through some reaction r implies zero flux $v_s = 0$, for some other reaction s.

Knockout analysis has become an important technique for the study of metabolic networks and in metabolic engineering. Starting from flux balance analysis (FBA), various *in silico* screening methods for genetic modifications have been developed, see [7,8] for an overview. On the one hand, complete methods have been proposed, which systematically explore all possible knockout sets up to a given size, e.g. [9,10]. On the other hand, there exist heuristic algorithms such as [11-14], which may be considerably faster, but in general are not complete. Klamt et al. [15-17] developed the related concept of *minimal cut sets*, which are (inclusion-wise) minimal sets of reactions whose knockout will block certain undesired flux distributions while maintaining others.

Recent progress in the development of algorithms for flux coupling analysis (FCA) [6,18] may lead to a different approach. FCA [5] describes the impact of each possible single reaction knockout in a metabolic network. It analyzes which other reactions become blocked after removing one reaction ("directional coupling"), and which reactions are always active together ("partial coupling"). As we will see, using flux coupling information inside a

*Correspondence: goldsteiny.de@gmail.com
FB Mathematik und Informatik, Freie Universität Berlin, Arnimallee 6, 14195, Berlin, Germany

double or multiple knockout simulation may significantly reduce the search space, without loosing any information.

In this paper, we present an algorithmic framework for double and multiple knockouts in qualitative models of metabolic networks. We will use a lattice-theoretic approach [18], which includes classical constraint-based models at steady-state as a special case, but which is much more general. We illustrate and evaluate our method by computing full double knockout simulations on a selection of genome-scale metabolic network reconstructions. In particular, we compare the impact of single vs. double reaction knockouts on the other reactions in the network. We also show how our method can be extended to gene (in contrast to reaction) knockouts, and provide computational results for both cases.

Our algorithms are based on an efficient search for the maximal element in suitably defined lattices [18]. To simulate all double or multiple reaction knockouts, we describe a method to select a subset of the reactions as representatives for the whole system. More precisely, we partition the reaction set in equivalence classes of *partially coupled* reactions. This enables us to obtain the information about all possible double or multiple reaction knockouts much faster and to store the results in a compact format.

The approach developed in this paper is a qualitative method. We do not measure the quantitative impact of knockout sets on the cellular growth rate (or other metabolic fluxes) as this would be done in an FBA approach. Instead, we count how many reactions become blocked by a knockout, similar to the *flux balance impact degree* introduced in [19]. However, even though we do not apply FBA to evaluate the impact of a knockout, the idea of working with representatives for reaction classes via partial coupling could also be applied in an FBA context. Thus, studies like [20] and even MILP-based approaches like [21] might benefit from this method.

Methods

Reaction coupling in the context of knockout analysis

We start from a metabolic network $\mathcal{N} = (\mathcal{M}, \mathcal{R}, S, \mathrm{Irr})$ given by a set of metabolites $\mathcal{M}$, a set of reactions $\mathcal{R}$, a stoichiometric matrix $S \in \mathbb{R}^{\mathcal{M}\times\mathcal{R}}$, and a set of irreversible reactions $\mathrm{Irr} \subseteq \mathcal{R}$, see Figure 1 for an example. The set $C = \{v \in \mathbb{R}^{\mathcal{R}} \mid Sv = 0, v_r \geq 0, r \in \mathrm{Irr}\}$ of all flux vectors $v \in \mathbb{R}^{\mathcal{R}}$ satisfying the steady-state (mass balance) constraints $Sv = 0$ and the thermodynamic irreversibility constraints $v_r \geq 0$, for all $r \in \mathrm{Irr}$, is called the *steady-state flux cone*. A reaction $s \in \mathcal{R}$ is called *blocked* if $v_s = 0$, for all $v \in C$, otherwise s is *unblocked*. Two unblocked reactions r, s are called *directionally coupled* [5], written $r \overset{=0}{\to} s$, if for all $v \in C$, $v_r = 0$ implies $v_s = 0$. A possible biological interpretation is that the reactions directionally coupled to r are those reactions that will become blocked by knocking out the reaction r.

To determine which reactions are coupled, a simple approach would be to solve for each pair of reactions (r, s) two linear programs (LPs) and to check whether $\max\{v_s \mid v \in C, v_r = 0\} = \min\{v_s \mid v \in C, v_r = 0\} = 0$. During the last years, efficient flux coupling algorithms have been developed [6,18] that drastically reduce the number of LPs to be solved, so that that genome-wide

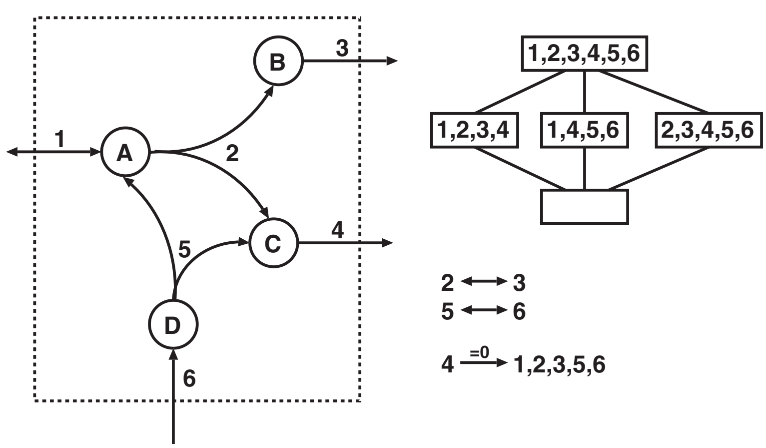

Figure 1 Example network with corresponding lattice and coupling relations. The network contains the set of metabolites $\mathcal{M} = \{A, B, C, D\}$ and the set of reactions $\mathcal{R} = \{1, 2, 3, 4, 5, 6\}$. We assume that all coefficients s_{mr} of the stoichiometric matrix S belong to $\{0, +1, -1\}$. Thus, reaction 2 has the stoichiometry $s_{A2} = -1, s_{B2} = s_{C2} = 1$ and $s_{D2} = 0$. The set of irreversible reactions is $\mathrm{Irr} = \mathcal{R} \setminus \{1\}$. A possible flux vector satisfying the steady-state condition $Sv = 0$ is $v = (0, 1, 1, 2, 1, 1)$ with supp $v = \{2, 3, 4, 5, 6\}$. The corresponding lattice contains the trivial element Ø representing the vector $v = 0$ and the minimal (non-trivial) elements $\{1, 2, 3, 4\}$, $\{1, 4, 5, 6\}$ and $\{2, 3, 4, 5, 6\}$. The maximal element is $\{1, 2, 3, 4, 5, 6\}$, i.e., there is no blocked reaction. a) There are two pairs of partially coupled reactions, namely $2 \leftrightarrow 3$ and $5 \leftrightarrow 6$. Therefore, no knockout sets containing reaction 3 or 5 need to be analysed. The impact of a double knockout of $\{3, r\}$ will be the same as for $\{2, r\}$. b) Reaction 1 is coupled to reaction 4. Thus, a double knockout of $\{1, 4\}$ will have the same effect as the simple knockout of 4. In both cases, all reactions $\{1, 2, 3, 4, 5, 6\}$ get blocked.

metabolic network reconstructions can now be analyzed in a few minutes on a desktop computer (compared to a couple of days of running time before).

Whether reactions are blocked or coupled does not depend on the specific flux values. It only matters whether or not $v_r = 0$ resp. $v_s = 0$. In this sense, flux coupling is a qualitative property that can be analysed by studying the set $L^C = \{\operatorname{supp} v \mid v \in C\}$ of all *supports* of flux vectors $v \in C$, where $\operatorname{supp} v = \{r \in \mathcal{R} \mid v_r \neq 0\}$. Each element $a \in L^C$ is the set of active reactions of some flux vector $v \in C$. Therefore, we can interpret L^C as the set of all *possible reaction sets* or *pathways* in the flux cone C. Since L^C does not contain any information about specific flux values, we also speak of a *qualitative model* of the metabolic network $\mathcal{N}$.

In [18,22], we have shown that flux coupling analysis can be extended to much more general qualitative models, where the space of possible pathways $L \subseteq 2^{\mathcal{R}}$ can be any non-empty subset of the power set $2^{\mathcal{R}}$, e.g. $L = \{\operatorname{supp} v \mid v \in C, v \text{ satisfies thermodynamic loop law constraints}\}$. The definition of flux coupling needs only be slightly modified in order to be applicable to these qualitative models. A reaction $t \in \mathcal{R}$ is called *blocked in L* if and only if for all $a \in L$, we have $t \notin a$. For reactions $r, s \in \mathcal{R}$ that are unblocked in L, we define $r \overset{=0}{\rightarrow} s$ in L, if for all $a \in L$, $r \notin a$ implies $s \notin a$. To distinguish between the original flux coupling and its qualitative extension, we will call the latter *reaction coupling* from now on.

The goal of this paper is to study more general dependencies between reactions, where the flux through some reaction has to be zero, if the flux through two or more other reactions is zero.

Definition 1 (Joint reaction coupling). *Given a qualitative model $L \subseteq 2^{\mathcal{R}}$ of a metabolic network $\mathcal{N}$, let $r, s, t \in \mathcal{R}$ be unblocked reactions in L such that neither $r \overset{=0}{\rightarrow} t$ in L nor $s \overset{=0}{\rightarrow} t$ in L holds. We say t is jointly coupled to the pair $\{r, s\}$ in L, written $\{r, s\} \overset{=0}{\rightarrow} t$ in L, if for all $a \in L$, $r \notin a$ and $s \notin a$ implies $t \notin a$.*

More generally, given a set $\mathcal{K} \subseteq \mathcal{R}$ of unblocked reactions in L, we say that t is jointly coupled to $\mathcal{K}$ in L, written $\mathcal{K} \overset{=0}{\rightarrow} t$ in L, if for all $a \in L$, $a \cap \mathcal{K} = \emptyset$ implies $t \notin a$, and $\mathcal{K}' \overset{=0}{\rightarrow} t$ in L does not hold for any $\emptyset \neq \mathcal{K}' \subsetneq \mathcal{K}$.

Note that in the definition of the joint coupling $\{r, s\} \overset{=0}{\rightarrow} t$ in L, we require that the simple couplings $r \overset{=0}{\rightarrow} t$ in L and $s \overset{=0}{\rightarrow} t$ in L both do *not* hold. Thus, *joint* coupling is about the synergistic effect of a pair of reactions r, s on some other reaction t, which cannot be obtained by either r or s alone. Similarly, $\mathcal{K} \overset{=0}{\rightarrow} t$ in L can only hold if $\mathcal{K}' \overset{=0}{\rightarrow} t$ in L does not hold, for any smaller knockout set $\emptyset \neq \mathcal{K}' \subsetneq \mathcal{K}$.

Lattices and maximal elements

In [18], we presented a generic algorithm for flux coupling analysis in qualitative models. This algorithm determines the pairs of coupled reactions by computing the maximal element in suitably defined lattices.

A family of reaction sets $L \subseteq 2^{\mathcal{R}}$ is a (finite) *lattice* if $\emptyset \in L$ and for all $a_1, a_2 \in L$, we have $a_1 \cup a_2 \in L$. The biological interpretation of this property is that the combination of two metabolic pathways is again a pathway. In [18] we showed that L^C is a lattice. Any finite lattice L has a unique *maximal element* 1_L (w.r.t. set inclusion), which is simply the union of all lattice elements, i.e., $1_L = \bigcup_{a \in L} a$. For any subset of reactions $\mathcal{K} \subseteq \mathcal{R}$, we may define the family

$$L_{\perp\mathcal{K}} = \{a \in L \mid a \cap \mathcal{K} = \emptyset\}$$

called L *without* $\mathcal{K}$ of those reaction sets $a \in L$ that do not contain any reaction in $\mathcal{K}$. If L is a lattice, then $L_{\perp\mathcal{K}}$ is a lattice again, and thus it has a maximal element

$$1_{L_{\perp\mathcal{K}}} = \bigcup_{a \in L, a \cap \mathcal{K} = \emptyset} a.$$

Given any lattice $L \subseteq 2^{\mathcal{R}}$, we have shown in [18] that a reaction $r \in \mathcal{R}$ is unblocked in L if and only if $r \in 1_L$. For two unblocked reactions $r, s \in 1_L$, the coupling relation $r \overset{=0}{\rightarrow} s$ in L holds if and only if $s \notin 1_{L_{\perp\{r\}}}$. In [18], we also presented an efficient algorithm to compute 1_L and $1_{L_{\perp\{r\}}}$. Once these maximal elements have been found, one can immediately determine the blocked and coupled reactions.

In this paper, we generalize these results to joint couplings. We present a method to compute the effects of double (resp. multiple) reaction knockouts based on the maximal element $1_{L_{\perp\{r,s\}}}$ (resp. $1_{L_{\perp\mathcal{K}}}$).

Proposition 1. *If $L \subseteq 2^{\mathcal{R}}$ is a lattice, then for any unblocked reactions $r, s, t \in 1_L$ we have*

$$\{r, s\} \overset{=0}{\rightarrow} t \text{ in } L \text{ if and only if } t \in \left(1_{L_{\perp\{r\}}} \cap 1_{L_{\perp\{s\}}}\right) \setminus 1_{L_{\perp\{r,s\}}}.$$

More generally, for a set of unblocked reactions $\mathcal{K} \subseteq 1_L$, we have

$$\mathcal{K} \overset{=0}{\rightarrow} t \text{ in } L \text{ if and only if } t \in \left(\bigcap_{k \in \mathcal{K}} 1_{L_{\perp\mathcal{K}\setminus\{k\}}}\right) \setminus 1_{L_{\perp\mathcal{K}}}.$$

Proof. We prove only the first part. The second part follows by induction.

Assume $\{r, s\} \overset{=0}{\rightarrow} t$ in L. By definition, we know $t \notin a$ for all $a \in L_{\perp\{r,s\}}$, and therefore $t \notin 1_{L_{\perp\{r,s\}}}$. If $\{r, s\} \overset{=0}{\rightarrow} t$ in L, we also know that neither $r \overset{=0}{\rightarrow} t$ in L nor $s \overset{=0}{\rightarrow} t$ in L and that all three reactions are unblocked, i.e., $r, s, t \in 1_L$. As discussed in [18], we have $r \overset{=0}{\rightarrow} t$ in L if and only if $t \in 1_L \setminus 1_{L_{\perp\{r\}}}$. Since $t \in 1_L$, we conclude $t \in 1_{L_{\perp\{r\}}}$, and by the same argument $t \in 1_{L_{\perp\{s\}}}$. Hence, $t \in \left(1_{L_{\perp\{r\}}} \cap 1_{L_{\perp\{s\}}}\right) \setminus 1_{L_{\perp\{r,s\}}}$.

If $t \in \left(1_{L_{\perp\{r\}}} \cap 1_{L_{\perp\{s\}}}\right) \setminus 1_{L_{\perp\{r,s\}}}$ holds, then $t \notin 1_{L_{\perp\{r,s\}}}$, which implies $t \notin a$ for all $a \in L_{\perp\{r,s\}}$. Since $t \in 1_{L_{\perp\{r\}}} \cap 1_{L_{\perp\{s\}}}$, we can again apply [18] to see that $r \overset{=0}{\rightarrow} t$ in L and $s \overset{=0}{\rightarrow} t$ in L do not hold. Finally, since $r, s, t \in 1_L$ are unblocked, we get $\{r, s\} \overset{=0}{\rightarrow} t$ in L. □

In [22], we considered even more general qualitative models $\emptyset \neq P \subseteq 2^{\mathcal{R}}$, where P needs not be a lattice. We showed there that qualitative flux coupling analysis can be done in the lattice $L^P = \langle P \rangle$ that is generated by P. The results we will present in this paper would be applicable to those qualitative models P as well, but for simplicity we will continue to work with models L that are lattices.

Classes of partially coupled reactions

To determine joint coupling relations $\mathcal{K} \overset{=0}{\rightarrow} t$ in L, we will use as much as possible the information that can be obtained from standard couplings $r \overset{=0}{\rightarrow} s$ in L, i.e., with normal FCA. If $r \overset{=0}{\rightarrow} s$ in L, any pathway $a \in L$ that does not use reaction r will also not use reaction s. Thus, knocking out s in addition to r will not affect the system, i.e., $\{a \in L \mid r, s \notin a\} = \{a \in L \mid r \notin a\}$.

Additional improvements can be obtained by looking at partially coupled reactions. Two unblocked reactions $r, s \in 1_L$ are called *partially coupled* in the lattice L, written $r \leftrightarrow s$, if both $r \overset{=0}{\rightarrow} s$ in L and $s \overset{=0}{\rightarrow} r$ in L. The relation $\leftrightarrow$ is reflexive, transitive and symmetric, and thus an equivalence relation. Any equivalence relation defines a partition of its ground set into equivalence classes. In our case, $1_L = \bigcup_{r \in 1_L} [r]_\leftrightarrow$, where $[r]_\leftrightarrow = \{s \in 1_L \mid r \leftrightarrow s\}$. An equivalence class can be represented by any of its elements, i.e., $[r]_\leftrightarrow = [\tilde{r}]_\leftrightarrow$ if $r \leftrightarrow \tilde{r}$. By selecting one element from each equivalence class, we get a set of *representatives* $\texttt{Rep} \subseteq 1_L$ that covers all unblocked reactions, i.e., $1_L = \bigcup_{r \in \texttt{Rep}} [r]_\leftrightarrow$. We will call $[r]_\leftrightarrow$ the *coupling class* or *reaction class* of reaction r. Biologically, coupling classes can be interpreted as subsets of reactions that are always active together, similarly to the notion of enzyme subsets in [23].

For $r, \tilde{r} \in [r]_\leftrightarrow$ and $a \in L$, we have $r \in a$ if and only if $\tilde{r} \in a$. Thus, a knockout of r has the same impact as a knockout of $\tilde{r}$. Furthermore, r can only be blocked by another knockout $k \notin [r]_\leftrightarrow$ if the same holds for $\tilde{r}$, i.e., $k \overset{=0}{\rightarrow} r$ in L if and only if $k \overset{=0}{\rightarrow} \tilde{r}$ in L. It follows that to analyse the effect of a knockout pair $\{\tilde{r}, \tilde{s}\}$, one can instead knockout the corresponding representatives $\{r, s\}$ with $\tilde{r} \in [r]_\leftrightarrow$ in L and $\tilde{s} \in [s]_\leftrightarrow$. To simulate all double knockouts, one does not have to check all pairs $\{\{\tilde{r}, \tilde{s}\} \mid \tilde{r}, \tilde{s} \in 1_L\}$, but it is enough to iterate over a fixed set of representatives: $\{\{r, s\} \mid r, s \in \texttt{Rep}\}$, see Figure 1a) for illustration. As we will see, for many genome-scale network reconstructions, there are only about half as many different equivalence classes as there are unblocked reactions (Table 1). Thus, only about 1/4 of all original pairs need to be checked. As mentioned before, although we apply this compression to reaction coupling analysis, it could also be combined with FBA-based methods.

Algorithms

In [18], we introduced an algorithm that performs flux coupling analysis by computing maximal elements of suitably defined finite lattices $\tilde{L}$ (see also the section above on lattices and maximal elements). The basic ingredient of this algorithm is a method that checks if a given reaction $r \in \mathcal{R}$ is blocked in $\tilde{L}$, and if not returns a pathway $a \in \tilde{L}$ with $r \in a$. The maximal element $1_{\tilde{L}}$ of $\tilde{L}$ is computed by improving lower and upper bounds $lb, ub \in \tilde{L}$ with $lb \subseteq 1_{\tilde{L}} \subseteq ub$. In each step of the algorithm, either lb is increased or ub is decreased, until finally $lb = ub = 1_{\tilde{L}}$. The following Algorithm 1 is an extension of this method. It allows finding all the reactions in $\mathcal{R}$ that are unblocked after a multiple knockout $\mathcal{K} \subseteq 1_L$.

Table 1 Knockout impact on different networks

Model			Single KOs	Double KOs	
	ub	classes	Impact	Impact	ratio
E. coli iJO1366	1718	1078	4.51 (16.6)	4.41 (10.1)	1.0%
E. coli iAF1260	1543	975	4.12 (13.7)	4.04 (9.24)	0.8%
S. cerevisiae iND750	631	371	5.42 (14.6)	5.52 (10.3)	2.7%
M. tuberculosis iNJ661	744	370	4.74 (35.6)	1.99 (5.78)	5.1%
S. aureus iSB619	465	207	11.7 (44.9)	7.31 (17.2)	9.2%
H. pylori iIT341	436	150	6.65 (58.6)	4.71 (15.5)	9.7%
E. coli textbook	87	55	1.96 (3.58)	15.7 (24.5)	12%

ub: Number of unblocked reactions in the original network.
classes: Number of different reaction classes, i.e., equivalence classes w.r.t. partial coupling $\leftrightarrow$.
Single KOs impact: Average impact of single reaction knockouts, i.e., average number of reactions classes that become blocked by a single knockout. In brackets: Average number of reactions that become blocked (belonging to different reaction classes).
Double KOs impact: Average *additional* impact of double reaction knockouts, i.e., average number of reactions classes that become blocked by a double knockout $\{r, s\}$, but are not blocked by a single knockout of either r or s. In brackets: Average number of additional reactions that become blocked.
Double KOs ratio: Percentage of pairs of (uncoupled) reaction classes that have joint coupling effects. The average numbers are determined by $\frac{1}{|K|}\sum_{\kappa \in K}$ impact (κ) with $K = \texttt{Rep}$ for the single, and $K = \{\{r, s\} \mid r, s \in \texttt{Rep}$ with neither $r \overset{=0}{\rightarrow} s$ in L nor $s \overset{=0}{\rightarrow} r$ for the double knockouts.

Algorithm 1. *Multiple Knockout Analysis*

Input: *A set of knockout reactions* $\mathcal{K} \subseteq 1_L$, $|\mathcal{K}| \geq 2$.
From FCA we reuse:

- *A set of representatives* $\texttt{Rep}$
- *Maximum elements* $1_{L_{\perp\{k\}}}$, *for* $k \in \mathcal{K}$
- *A set of previously computed pathways* $\mathcal{W} \subseteq L$ *(witnesses)*

Ouput: *The set of reactions* $1_{L_{\perp\mathcal{K}}}$ *that are unblocked in the subnetwork* $\mathcal{R} \setminus \mathcal{K}$.

function MKO *(*$\mathcal{K}$*)*
  $\texttt{lb} = \bigcup_{a \in \mathcal{W}_{\perp\mathcal{K}}} a$, *with* $\mathcal{W}_{\perp\mathcal{K}} = \{a \in \mathcal{W} \mid a \cap \mathcal{K} = \emptyset\}$
  $\texttt{ub} = \bigcap_{k \in \mathcal{K}} 1_{L_{\perp\{k\}}}$
  for $r \in \texttt{Rep}$ ***do***
    if $r \in \texttt{ub} \setminus \texttt{lb}$ ***then***
      $a = \text{FindPath}\,(r, \mathcal{K})$
      if $r \in a$ ***then***
        $\texttt{lb} = a \cup \texttt{lb}$
      else
        $\texttt{ub} = \texttt{ub} \cap 1_{L_{\perp\{r\}}}$
  return $1_{L_{\perp\mathcal{K}}} = \texttt{ub}$

function FindPath *(*$r, \mathcal{K}$*)*

$$\textbf{\textit{return}} \begin{cases} a & \text{there exists } a \in L : r \in a,\ a \cap \mathcal{K} = \emptyset, \\ \emptyset & \text{otherwise.} \end{cases}$$

Algorithm 2. *Full Double Knockout Analysis*

Input: *From FCA we reuse:*

- *A set of representatives* $\texttt{Rep}$
- *Maximum elements* $1_{L_{\perp\{r\}}}$, *for* $r \in \texttt{Rep}$
- *A set of previously calculated pathways* $\mathcal{W} \subseteq L$ *(witnesses)*

Output: *The set* $\texttt{dkos}$ *containing all joint couplings* $\{r, s\} \overset{=0}{\to} t$ *in* L, *with* $r, s, t \in \texttt{Rep}$.

$\texttt{dkos} = \emptyset$
for $r, s \in \texttt{Rep}$ *with* $r < s$ ***do***
  if $r \in 1_{L_{\perp\{s\}}}$ *and* $s \in 1_{L_{\perp\{r\}}}$ ***then***
    $\texttt{lb} = \bigcup_{a \in \mathcal{W}_{\perp\{r,s\}}} a$, *with* $\mathcal{W}_{\perp\{r,s\}} = \{a \in \mathcal{W} \mid r, s \notin a\}$
    $\texttt{ub} = \texttt{known} = 1_{L_{\perp\{r\}}} \cap 1_{L_{\perp\{s\}}}$
    for $t \in \texttt{Rep}$ ***do***
      if $t \in \texttt{ub} \setminus \texttt{lb}$ ***then***
        $a = \text{FindPath}\,(t, \{r, s\})$
        if $t \in a$ ***then***
          $\texttt{lb} = a \cup \texttt{lb}$
        else
          $\texttt{ub} = \texttt{ub} \cap 1_{L_{\perp\{t\}}}$
  $\texttt{dkos} = \texttt{dkos} \cup \left\{\{r, s\} \overset{=0}{\to} t \text{ in } L \mid t \in \texttt{known} \setminus \texttt{ub}\right\}$
  return $\texttt{dkos}$

As discussed in [18], the flexibility of the lattice-based approach comes from hiding the search for specific pathways in a separate function $\texttt{FindPath}$. For traditional steady-state based models, $\texttt{FindPath}$ can be realized by solving the linear programs $\max\{\pm v_t | Sv = 0, v_{\text{Irr}} \geq 0, v_k = 0, k \in \mathcal{K}\}$. But, one can also use other modeling hypotheses and corresponding algorithmic methods (see [22] for the example of thermodynamic loop law constraints). The skeleton of Algorithm 1 will remain the same, only the auxiliary function $\texttt{FindPath}$ has to be changed.

In Algorithm 1, we perform a multiple knockout analysis with a fixed knockout set $\mathcal{K}$. For a full d-dimensional knockout analysis, we would have to iterate over all $\mathcal{K} \subseteq 1_L$ with $|\mathcal{K}| = d$, i.e., we would have to run the algorithm $O\left(\binom{|\mathcal{R}|}{d}\right)$ times. In each iteration, we have to solve $O(|\mathcal{R}|)$ linear programs. Since linear programming can be done in polynomial time, full d-dimensional knockout analysis is still polynomial (for fixed d), but computationally very expensive as soon as $d > 2$. However, we can still use the partition of 1_L into equivalence classes of partially coupled reactions. Thus, our next Algorithm 2 calculates representatives of all jointly coupled reactions in the case of double knockouts.

In Algorithm 2, we iterate over a subset of all possible double knockouts without loosing any information. For this, we filter redundant knockout pairs such as $r \overset{=0}{\to} s$ in L (by checking $s \in 1_{L_{\perp\{r\}}}$). It is unnecessary to test such a pair, because a knockout of $\{r, s\}$ is equivalent to the single knockout of r, see Figure 1b) for illustration. For higher-dimensional knockout sets one can proceed in a similar fashion:

Let $\mathcal{K} = \{k_1, \ldots, k_d\} \subseteq \texttt{Rep}$ be a d-dimensional knockout set. Then we do *not* need to test $\mathcal{K}$, if any of the following conditions is fulfilled:

- $k_i \overset{=0}{\to} k_j$ in L for two reactions $k_i, k_j \in \mathcal{K}$,
- $\{k_{i_1}, k_{i_2}\} \overset{=0}{\to} k_j$ in L for three reactions $k_{i_1}, k_{i_2}, k_j \in \mathcal{K}$,
- $\{k_{i_1}, k_{i_2}, k_{i_3}\} \overset{=0}{\to} k_j$ in L for four reactions $k_{i_1}, k_{i_2}, k_{i_3}, k_j \in \mathcal{K}$,
- etc.

Standard FCA finds all pairs of reactions that are directionally coupled. This allows us to iterate in Algorithm 2 over all $\{r, s\} \in \mathcal{K}_{2,1}$ with

$$\mathcal{K}_{2,1} = \left\{\{k_1, k_2\} \subseteq \texttt{Rep} \mid \text{not} k_1 \overset{=0}{\to} k_2 \text{ in } L \text{ and not } k_2 \overset{=0}{\to} k_1 \text{ in } L\right\}.$$

$\mathcal{K}_{2,1}$ contains all 2-tuples of coupling class representatives that are not coupled with respect to knockouts up to cardinality 1.

If one is interested to perform a full triple knockout analysis and joint coupling information is available, one can adapt the filtering technique and iterate over all $\{r_1, r_2, r_3\} \in \mathcal{K}_{3,1}$ (or $\mathcal{K}_{3,2}$) with

$$\mathcal{K}_{3,1} = \Big\{\{k_1, k_2, k_3\} \subseteq \mathtt{Rep} \mid \text{not } k_i \stackrel{=0}{\to} k_j \text{ in } L, \text{ for all } i \neq j \in \{1,2,3\}\Big\},$$

$$\mathcal{K}_{3,2} = \Big\{\{k_1, k_2, k_3\} \subseteq \mathtt{Rep} \mid \text{not } k_{i_1} \stackrel{=0}{\to} k_j \text{ in } L \text{ and not } \{k_{i_1}, k_{i_2}\} \stackrel{=0}{\to} k_j \text{ in } L, \text{ for all pairwise different } i_1, i_2, j \in \{1,2,3\}\Big\}.$$

$\mathcal{K}_{3,1}$ contains all 3-tuples of coupling class representatives that are not directionally coupled, and $\mathcal{K}_{3,2}$ all triples that do not contain reactions that are coupled with respect to knockouts up to cardinality 2. Similarly one could define $\mathcal{K}_{d,m}$.

While these techniques are applied here only to reaction coupling analysis, they could also be combined with FBA-based methods. Thus, if one is interested to measure the impact of all possible triple knockouts on FBA, it would be sufficient to solve $\max\{v_{biomass} | Sv = 0, v_{\mathrm{Irr}} \geq 0, v_{\mathcal{K}} = 0\}$ for all $\mathcal{K} \in \mathcal{K}_{3,1}$ (if only FCA data is available) or all $\mathcal{K} \in \mathcal{K}_{3,2}$ (if FCA and joint coupling data is available).

The case of gene knockouts

Often metabolic networks contain regulatory rules for the gene products that catalyze the reactions, e.g. reaction r_1 is catalyzed by the product of a gene g_1 and reaction r_2 is catalyzed by the gene product of g_1 or g_2. Here r_1 is only possible if g_1 is active, and r_2 can only be blocked by a simultaneous knockout of the two genes g_1 and g_2. Typically, there is no 1-1 relationship between the set of genes $\mathcal{G}$ and the set of reactions $\mathcal{R}$. On the one hand, there are reactions that only get blocked by a combination of two or more gene knockouts, as indicated above in $r_2 \equiv g_1 \vee g_2$. On the other hand, the knockout of a single gene $g \in \mathcal{G}$ may block more than one reaction. For example, reactions r_1 and r_3 may both depend on the gene g_1. Then one immediately gets that a knockout of g_1 implies $v_1 = v_3 = 0$. Let us further assume that FCA and double reaction knockout analysis have been performed, leading to $3 \stackrel{=0}{\to} 4$ in L and $\{1,3\} \stackrel{=0}{\to} 6$ in L. Based on this information, we can extend the reactions that are blocked by the knockout of gene g_1 to $v_1 = v_3 = v_4 = v_6 = 0$. Thus, in this example we have 2 reactions (r_1, r_3) that are *associated to the gene* g_1 based on information that is directly available in the network reconstruction, but in total 4 reactions (r_1, r_3, r_4, r_6) that are *coupled to the gene* g_1. We formalize these notions in the following definition.

Definition 2 (Gene coupling)**.** *Consider a qualitative model $L \subseteq 2^{\mathcal{R}}$ of a metabolic network $\mathcal{N}$ with reaction set $\mathcal{R}$ and gene set $\mathcal{G}$. Let $\alpha : 2^{\mathcal{G}} \to 2^{\mathcal{R}}, \Gamma \mapsto \mathcal{K}_\Gamma$ be a function defining a set of reactions $\mathcal{K}_\Gamma$ associated to the knockout of all genes in the set Γ. For an unblocked reaction $r \in 1_L$ and $\Gamma \subseteq \mathcal{G}$ we define:*

$$\Gamma \stackrel{=0}{\to} r \text{ in } L \text{ if and only if } r \notin 1_{L_{\perp \mathcal{K}_\Gamma}}.$$

We say that the reaction r is coupled to the gene knockout Γ. If $\Gamma = \{g\}$ is a single gene, we simply write $g \stackrel{=0}{\to} r$ in L.

Given the function $\alpha : 2^{\mathcal{G}} \to 2^{\mathcal{R}}$, we can determine the reactions coupled to the gene set Γ by applying Algorithm 1 to the set of associated reactions $\mathcal{K}_\Gamma$. Note that the definition of gene coupling slightly differs from the one of joint reaction coupling. Here, we do not exclude reactions that are already knocked out by single (or smaller set of) gene knockouts. This is to account for the possibility that, for example, a reaction r may be associated to a single gene knockout g_1, but not to the double knockout $\{g_1, g_2\}$ (assume $r \equiv g_1 \vee \neg g_2$).

To simulate the impact of all single gene knockouts, one can perform an iteration over all genes $g \in \mathcal{G}$. Similarly, one can determine all double gene knockout effects by an iteration over all pairs of genes $\{g_1, g_2\} \subseteq \mathcal{G}$. However, in contrast to Algorithm 2, we cannot use gene class representatives to decrease the number of pairs that have to be analyzed.

Results and discussion

To evaluate our method, we simulated all single and double reaction knockouts for a number of genome-scale metabolic network reconstructions from the BiGG-database [24]. The computations were done on a MacBook Air (2012), with 1.8 GHz Intel Core i5, 4GB RAM, and running `Java Oracle JDK` 1.7.45 under `Mac OS X` 10.9. To solve linear programs (LPs), we used `CPLEX` Version 12.6.

Impact of double knockouts

Table 1 shows the impact of single and double reaction knockouts for the different networks. In most cases, the knockout of a single reaction class (due to the knockout of one or more of its reactions) blocks the reactions in 4 to 5 other reaction classes in average. The least robust system is *S. aureus* iSB619, where a single knockout has an average impact of almost 12 coupled reaction classes. In *S. aureus* iSB619, about 9.2% of all possible double knockouts $\{r, s\}$ have *joint* coupling effects, i.e., there exist reactions $t \in \mathcal{R}$ that are blocked by the double knockout $\{r, s\}$, but not by a single knockout of r or s alone. This is a comparatively large number. For the bigger *E. coli* models iAF1260 and iJO1366, only around 1% of all double knockouts of two uncoupled reaction classes $\{r, s\}$ have an impact that exceeds the effects of the corresponding two single knockouts. In *S. aureus* iSB619, double knockouts also have very strong combined effects. In addition to the reaction classes that would be knocked out by r or s alone, in average more than 7 reaction classes are coupled to a double knockout corresponding to a joint coupling $\{r, s\} \overset{=0}{\rightarrow} t$ in L. But, even for the most robust system, *M. tuberculosis* iNJ661, a double knockout (if its impact is different from the two single knockouts) in average has a combined effect of 2 additional knocked out classes resp. 5.8 reactions.

Knockout options

In our next experiment, we take the opposite perspective (Table 2). We analyse how robust an average reaction is to single or double knockouts. More precisely, we ask the following question: Given a reaction t, what are the possible choices for a single reaction r resp. a pair of reactions $\{r, s\}$ such that $r \overset{=0}{\rightarrow} t$ in L resp. $\{r, s\} \overset{=0}{\rightarrow} t$ in L holds. This perspective corresponds to a lab experiment for finding knockout targets for the reaction t. Here, we consider single reactions instead of reaction classes. This means that for $\{r, s\} \overset{=0}{\rightarrow} t$ in L with $r, s, t \in \texttt{Rep}$, we get $|[r]| \cdot |[s]|$ knockout options for all the $|[t]|$ reactions that belong to the same reaction class as t.

For most of the studied networks, the average number of knockout options for a given target reaction is in the range of 25-85 single reactions and 100-150 reaction pairs. With all double knockout information at hand, one can reduce the set of all possible knockout candidates for a wet lab experiment to a small number, and additionally decide beforehand which of them have the smallest side effects.

Table 2 Average number of knockout options

Model	Single KOs	Double KOs
	Options	Options
E. coli iJO1366	35.1	143
E. coli iAF1260	26.4	78.0
S. cerevisiae iND750	25.6	106
M. tuberculosis iNJ661	82.7	120
S. aureus iSB619	65.9	245
H. pylori iIT341	143	126
E. coli textbook	6.92	132

Single KOs options: Average number of reactions r that lead as single knockouts to inactivity of a target reaction t: $\frac{1}{|1_L|} \sum_{t \in 1_L} \sum_{r \overset{=0}{\rightarrow} t \text{ in } L} 1$.
Double KOs options: Average number of uncoupled reaction pairs $\{r, s\}$ that lead as double knockouts to inactivity of a target reaction t: $\frac{1}{|1_L|} \sum_{t \in 1_L} \sum_{\{r,s\} \overset{=0}{\rightarrow} t \text{ in } L} 1$.

Impact on biomass production

To finish our discussion, we study the impact of knockouts on biomass production. To measure this, we counted the number of single and double knockouts that block the biomass reaction. Table 3 presents the results for the largest available models of the respective organisms. For two of them, more than one biomass reaction was available. In the case of *E. coli* iJO1366, we present the results for the two biomass reactions, for *S. aureus*, we selected 2 out of the 14 available reactions.

We observe that for most of the organisms, the number of single knockouts that block biomass production is very similar to the number of different double knockouts (corresponding to joint couplings) having this property, although the number of double knockout candidates is much larger (quadratic in $|1_L|$).

Algorithmic considerations

To perform a double knockout analysis, we first run standard flux coupling analysis (FCA) using the `L4FC` routine from [18]. Then we calculate the unblocked reactions for each double knockout of a pair of reaction class representatives. Table 4 presents the running times for six genome-scale network reconstructions and the central metabolism of *E. coli*. Even for our largest network, *E. coli* iJO1366 with its 2583 reactions, the complete simulation of all double reaction knockouts took less than 1h 10 min.

Next we discuss the number of LPs we have to solve in order to obtain this additional information. For all our networks, double knockout analysis required solving 5 to 20 times as many LPs than single knockouts, i.e., classical FCA. While this seems to be a large number, it is relatively small compared to the complexity of the problem. A full double knockout simulation is comparable to iterating over all reactions $r \in \texttt{Rep}$, removing the reaction r and performing a single knockout simulation for each of the resulting subnetworks. Reusing known pathways as witnesses and including reaction coupling information as proposed in [18] allows performing $|\texttt{Rep}|$ simulations with only 5 to 20 times the effort in LP solving. Table 1 shows that the median value for $|\texttt{Rep}|$ is 370 for our networks.

In order to evaluate the runtime effect of our algorithmic improvements, we considered two variants of Algorithm 2:

Variant A (no representatives) In the main loop of Algorithm 2, we do not iterate over all representatives $r, s \in \texttt{Rep}, r < s$, but over all pairs of uncoupled reactions $r, s \in 1_L, r < s$, with not $r \overset{=0}{\rightarrow} s$ *in* L and not $s \overset{=0}{\rightarrow} r$ in L.

Variant B (no witnesses) Same as Variant A. Additionally we do not save witnesses, thus $\mathcal{W} = \emptyset$.

Table 3 Number of knockouts for the biomass reaction in selected networks

Model		Single knockouts		Double knockouts	
reaction id	cl. size	classes	reactions	cl. pairs	reac. pairs
E. coli iJO1366					
`Ec_biomass_iJO1366_WT_53p95M`	20	101	343	130	339
`Ec_biomass_iJO1366_core_53p95M`	1	80	288	90	268
S. cerevisiae iND750					
`biomass_SC4_bal`	26	54	156	60	142
M. tuberculosis iNJ661					
`biomass_Mtb_9_60atp`	160	64	154	48	83
S. aureus iSB619					
`SA_biomass_1a`	8	25	63	59	157
`SA_biomass_5a`	1	58	215	54	100
H. pylori iIT341					
`BiomassHP_published`	189	36	76	41	81

class size: Number of reactions in the same coupling class as the biomass reaction, i.e., number of reactions that carry flux if and only if the biomass reaction carries flux.
Single Knockouts: Number of different single knockouts (classes and reactions) that block the biomass reaction. Only reactions that are not partially coupled to the biomass (from a different reaction class) are counted.
Double Knockouts: Number of different double knockouts (class pairs and reaction pairs) that block the biomass reaction when combined. Only reactions that are not directionally coupled to the biomass are counted.

These two experiments allow determining time savings due to representatives (comparing Algorithm 2 and Variant A) and time savings due to warm starts based on knowledge of existing reaction sets (comparing Variant A and B). We should emphasize here that Variant B is is still more efficient than a naive brute force algorithm. The runtime results are given in Table 5, where we stopped computations after a timeout of 6h. Table 5 shows that the efficiency of Algorithm 2 is mostly due to the re-use of (up to 10000) pathways as witnesses (factor 10 in the case of *E. coli textbook* and factor 100 for *S. aureus*). Nevertheless, iterating over the set of representatives adds another improvement of up to 50% (*S. aureus*). Since it takes a very small effort to calculate a set of representatives to profit from this additional speed-up, we highly recommend to iterate over representatives whenever possible.

Table 4 Runtime and number of solved LPs for double *reaction* knockouts (Algorithm 2)

Model		Step			Total
		Blocked	Couples	dko	
E. coli iJO1366	LPs	1718	9943	133225	144886
	time	2.0	42.2	4016.4	1h 8min
E. coli iAF1260	LPs	1679	10780	52112	64571
	time	1.7	31.5	2688.2	45min 21s
S. cerevisiae iND750	LPs	597	3987	90664	95248
	time	0.33	6.8	397.8	6min 45s
M. tuberculosis iNJ661	LPs	327	3416	20647	24390
	time	0.33	5.6	177.7	3min 4s
S. aureus iSB619	LPs	144	3638	19477	23259
	time	0.09	2.8	43.2	46.0s
H. pylori iIT341	LPs	106	1812	6753	8671
	time	0.06	1.9	18.0	20.0s
E. coli textbook	LPs	26	341	1739	2106
	time	0.004	0.06	0.62	0.68s

The computation was done in three steps: Calculation of the blocked reactions, flux coupling analysis to determine the coupled reactions, and finally the double knockout simulations.
Times are given in seconds if not specified otherwise (numbers may not add up due to rounding errors).

Gene knockouts

Table 6 gives the runtimes and the number of LPs for single and double *gene* knockouts. To determine the reactions associated to a (double) gene knockout, we used the library `JEval` that allows fast evaluation of logical

Table 5 Runtime of variants of Algorithm 2 for computing double reaction knockouts

Model	Algor. 2	Variant A (no representatives)	Variant B (no witnesses)
E. coli iJO1366	1h 8min	1h 59min	> 6h
E. coli iAF1260	45min 21s	1h 30min	> 6h
S. cerevisiae iND750	6min 45s	8min 39s	> 6h
M. tuberculosis iNJ661	3min 4s	7min 42s	> 6h
S. aureus iSB619	46s	1min 59s	2h 32min
H. pylori iIT341	20s	58.1s	52min
E. coli textbook	0.68s	2.2s	23.0s

Table 6 Runtime and number of solved LPs for single and double *gene* knockouts

Model		Step	
		gko	dgko
E. coli iJO1366	LPs	719	263844
	time	1.2	3h 49min
E. coli iAF1260	LPs	516	229498
	time	8.6	2h 55min
S. cerevisiae iND750	LPs	1323	308145
	time	6.4	37min 36s
M. tuberculosis iNJ661	LPs	175	77346
	time	1.2	15min 59s
S. aureus iSB619	LPs	49	38689
	time	0.68	9min 42s
H. pylori iIT341	LPs	27	19348
	time	0.24	1min 52s
E. coli textbook	LPs	2	2023
	time	0.04	4.4s

Times are given in seconds if not specified otherwise.

formulas given as Java strings. As expected we are confronted with longer runtimes up to almost 4h for double gene knockouts compared to < 70 min for double reaction knockouts. This is due to the fact that we need to check every single pair of genes instead of a representative selection like the one we could apply in double reaction knockout analysis. In spite of this, with the methods proposed here, a full simulation of double reaction or double gene knockouts on a genome-scale metabolic network reconstruction can still be performed in a reasonable time.

Conclusions

On the algorithmic side, this study presented the following main results:

- Algorithm 2 is an effective method for a complete double knockout analysis in genome-scale metabolic networks.
- Using Algorithm 1, it is possible to compute the impact of specific multiple knockout sets containing 3 or more reactions.
- By exploiting the information present in reaction coupling data (obtained by FCA), one can significantly decrease the number of candidates that need to be tested in double and multiple knockout simulations.

Regarding the biological data, we can make the following observations based on our computational experiments:

- In the genome-scale metabolic network reconstructions that were considered in this study, 1-10% of the possible double knockout sets have joint coupling effects. Thus, given a randomly chosen reaction pair, the probability is high that the combined effect of the double knockout (in terms of other blocked reactions) will be the same as for the two corresponding single knockouts.
- However, in all these networks, there exists a small number of double knockouts showing synergistic effects, blocking 5 to 20 additional reactions in average. These double knockouts cannot be predicted from the single knockout/reaction coupling data alone.

Due to the algorithmic improvements, we are now able to perform full double gene or reaction knockout simulations in a few hours of computation time. Thus, whenever one is interested in understanding the robustness of a network to knockouts, one should take the opportunity and run such an *in silico* simulation, before starting other more time consuming and expensive experiments.

A prototype implementation of double knockout simulation is available at http://hoverboard.io/L4FC.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The paper is based on the PhD thesis of YG, which was supervised by AB. YG implemented the algorithms and performed the computational experiments. YG and AB together wrote the manuscript and approved the final version.

Acknowledgements

The PhD work of Yaron Goldstein was supported by the Berlin Mathematical School and the Gerhard C. Starck Stiftung.

Received: 30 March 2014 Accepted: 4 December 2014
Published online: 09 January 2015

References

1. Bordbar A, Monk JM, King ZA, Palsson B. Constraint-based models predict metabolic and associated cellular functions. Nat Rev Genet 2014;15(2):107–120.
2. Lewis NE, Nagarajan H, Palsson B. Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods. Nat Rev Microbiol 2012;10(4):291–305.
3. Varma A, Palsson BO. Predictions for oxygen supply control to enhance population stability of engineered production strains. Biotechnol Bioeng 1994;43(4):275–285.
4. Orth JD, Thiele I, Palsson BO. What is flux balance analysis?. Nat Biotechnol 2010;28(3):245–248.
5. Burgard AP, Nikolaev EV, Schilling CH, Maranas CD. Flux coupling analysis of genome-scale metabolic network reconstructions. Genome Res 2004;14(2):301–312.
6. Larhlimi A, David L, Selbig J, Bockmayr A. F2C2: a fast tool for the computation of flux coupling in genome-scale metabolic networks. BMC Bioinformatics 2012;13(1):57.
7. Tomar N, De RK. Comparing methods for metabolic network analysis and an application to metabolic engineering. Gene 2013;521(1):1–14.
8. Zomorrodi AR, Suthers PF, Ranganathan S, Maranas CD. Mathematical optimization applications in metabolic networks. Metabolic Eng 2012;14(6):672–686.
9. Burgard AP, Pharkya P, Maranas CD. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol Bioeng 2003;84(6):647–657.

10. Tepper N, Shlomi T. Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. Bioinformatics 2010;26(4):536–543.
11. Patil KR, Rocha I, Förster J, Nielsen J. Evolutionary programming as a platform for in silico metabolic engineering. BMC Bioinformatics 2005;6(1):308.
12. Lun DS, Rockwell G, Guido NJ, Baym M, Kelner JA, Berger B, Galagan JE, Church GM. Large-scale identification of genetic design strategies using local search. Mol Syst Biol 2009;5(1): 296.
13. Rocha I, Maia P, Evangelista P, Vilaça P, Soares S, Pinto JP, Nielsen J, Patil KR, Ferreira EC, Rocha M. Optflux: an open-source software platform for in silico metabolic engineering. BMC Syst Biol 2010;4(1):45.
14. Ohno S, Shimizu H, Furusawa C. FastPros screening of reaction knockout strategies for metabolic engineering. Bioinformatics 2014;30(7):981–987.
15. Klamt S, Gilles ED. Minimal cut sets in biochemical reaction networks. Bioinformatics 2004;20(2):226–234.
16. Jungreuthmayer C, Nair G, Klamt S, Zanghellini J. Comparison and improvement of algorithms for computing minimal cut sets. BMC Bioinformatics 2013;14(1):318.
17. von Kamp A, Klamt S. Enumeration of smallest intervention strategies in genome-scale metabolic networks. PLOS Comput Biol 2014;10(1): 1003378.
18. Goldstein YAB, Bockmayr A. A lattice-theoretic framework for metabolic pathway analysis In: Gupta A, Henzinger T, editors. Computational Methods in Systems Biology. Lecture Notes in Computer Science. Vol. 8130, Berlin: Springer; 2013. p. 178–191.
19. Zhao Y, Tamura T, Akutsu T, Vert J-P. Flux balance impact degree: a new definition of impact degree to properly treat reversible reactions in metabolic networks. Bioinformatics 2013;29(17):2178–2185.
20. Nogales J, Gudmundsson S, Thiele I. An in silico re-design of the metabolism in thermotoga maritima for increased biohydrogen production. Int J Hydrogen Energy 2012;37(17):12205–12218.
21. Suthers PF, Zomorrodi A, Maranas CD. Genome-scale gene/reaction essentiality and synthetic lethality analysis. Mol Syst Biol 2009;5(1):301.
22. Reimers AC, Goldstein YAB, Bockmayr A. Qualitative and thermodynamic flux coupling analysis. Technical Report #1054, Matheon (March 2014). http://nbn-resolving.de/urn:nbn:de:0296-matheon-12801
23. Pfeiffer T, Sánchez-Valdenebro I, Nuño JC, Montero F, Schuster S. METATOOL: for studying metabolic networks. Bioinformatics 1999;15: 251–257.
24. Schellenberger J, Park JO, Conrad TM, Palsson BO. BiGG: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. BMC Bioinformatics 2010;11(213):213.