

# Enumerating precursor sets of target metabolites in a metabolic network

L. Cottret<sup>1,2</sup>, P. V. Milreu<sup>4</sup>, V. Acuña<sup>1,2</sup>, A. Marchetti-Spaccamela<sup>3</sup>,  
F. Viduani Martinez<sup>4</sup>, M.-F. Sagot<sup>1,2</sup>, and L. Stougie<sup>5</sup>

<sup>1</sup> Université de Lyon, F-69000, Lyon ; Université Lyon 1 ; CNRS, UMR5558  
<sup>2</sup> Projet Helix, INRIA Rhône-Alpes, France,  
{cottret,viacuna}@biomserv.univ-lyon1.fr, marie-france.sagot@inria.fr  
<sup>3</sup> Sapienza University of Rome, alberto.marchetti@dis.uniroma1.it  
<sup>4</sup> Universidade Federal de Mato Grosso do Sul,  
fhvm@dct.ufms.br, paulovieira@milreu.com.br  
<sup>5</sup> Eindhoven University of Technology and CWI, Amsterdam,leen@win.tue.nl

**Abstract.** We present the first exact method based on the topology of a metabolic network to find minimal sets of metabolites (called precursors) sufficient to produce a set of target metabolites. In contrast with previous proposals, our model takes into account self-regenerating metabolites involved in cycles, which may be used to generate target metabolites from potential precursors. We analyse the complexity of the problem and we propose an algorithm to enumerate all minimal precursor sets for a set of target metabolites. The algorithm can be applied to identify a minimal medium necessary for a cell to ensure some metabolic functions. It can be used also to check inconsistencies caused by misannotations in a metabolic network. We present two illustrations of these applications.

## 1 Introduction

The metabolic capacities of an organism are directly defined by the set of its possible biochemical reactions. The links between reactions and compounds (or metabolites) that are used/produced by such reactions constitute the *metabolic network* of an organism. Once the metabolic network of an organism has been defined (see [2] for an overview of the metabolic data reconstruction process), the following important question arises: how are the essential metabolites for the organism produced? Equivalently, which are the metabolites that the organism needs to obtain from its environment to produce those essential metabolites? In the sequel, we call such metabolites *precursors*.

One way to answer this question is to manually inspect the metabolic pathways defined as present in the organism: the presence of any metabolic pathway is determined by comparing the set of reactions of a reconstructed metabolic network with the set of reactions in the reference metabolic pathways contained in metabolic databases such as METACYC from the BIOCYC database collection [1] or KEGG [5]. However, each reference metabolic pathway represents a very small

part of the whole network and does not consider what occurs upstream of the pathway, nor whether some alternative organism-specific pathways exist.

With the goal of detecting inconsistencies in ECOCYC (the pathway-genome database dedicated to the bacterium *Escherichia coli*), Romero and Karp [8] used a whole-network approach to find precursors, while Handorf *et al.* [4] proposed a method to identify minimal metabolite sets required by an organism to produce all metabolites contained in a target set. In the method of Romero and Karp, the definition of potential precursors is very restrictive while it is very broad in Handorf *et al.*'s method. In this paper, we propose an exact method that may deal with any set of potential precursors defined by the user. Our method also takes into account the fact that most reactions are defined as reversible because of a lack of information on metabolite concentrations and enzyme kinetic properties. It can be used with a directed, undirected or mixed hypergraph representation of a metabolic network. It is not clear whether previous proposals could handle such cases.

In their paper, Romero and Karp did also not provide any details on how they dealt with cycles of reactions, a crucial issue when analysing metabolic networks. A similar consideration applies to the Handorf method, where a reaction can be fired only if all the metabolites are in the sub-network already produced by the process. The method is therefore not able to take into account metabolites which cannot be reached by such a process. The second main contribution of this paper is to address this problem by explicitly dealing with cycles when computing precursor sets. By a *cycle*, we refer to the concept of cycle in a hypergraph representation of a metabolic network which we describe in detail in Section 2.

This calls for the introduction of the new, biologically well-founded concept of “self-regenerating” metabolites that cannot be considered as available in infinite supply, e.g. provided by the environment, as is assumed to be the case for precursors. Such metabolites need to be continuously regenerated, but they have the ability to participate in their own regeneration, and in the subsequent generation of other metabolites. Self-regenerating metabolites will be part of at least one cycle. In [8], Romero and Karp very informally define what they call *bootstrapping* compounds that may be related to our self-regenerating metabolites but only partially and the list of such compounds needs furthermore to be provided as input by the user.

In Section 2, we give basic notations and definitions. In Section 3, we analyse the complexity of finding a minimal and a minimum precursor set. An exact method for enumerating all minimal precursor sets for a given set of target metabolites is described in Section 4. Finally, the method is illustrated with two applications in Section 5. For the sake of space most proofs are omitted.

## 2 Preliminaries

A metabolic network consists of a set of metabolites and a set of reactions. Each reaction transforms a subset of metabolites, the *substrates*, into another subset of metabolites, the *products* of the reaction. Such a network can be mod-

eled as a directed hypergraph  $G = (\mathcal{C}, \mathcal{R})$  with  $\mathcal{C}$  the set of vertices corresponding to *metabolites* (also called *compounds*) and  $\mathcal{R}$  the set of hyperedges corresponding to *reactions*. A hyperedge  $r \in \mathcal{R}$  is directed away from a compound  $c \in \mathcal{C}$  only if  $c$  is a substrate of  $r$ , and directed into  $c$  only if  $c$  is a product of  $r$ . Note that reactions can be reversible: each reversible reaction is modelled as two different reactions of opposite direction.

A solution to the problem of finding precursors for a specific set of target metabolites is a set of compounds that are in “infinite” supply (for instance, from the environment) and can be used as substrates of some reactions. These reactions will then produce new metabolites, thereby increasing the set of available metabolites. By iterating the process, we can check whether the target is produced.

This way of considering the dynamics of a network is not enough to model the real process. Indeed, the network could have cycles that regenerate their own metabolites: metabolites that are not available initially could still be used as substrates of reactions because they are part of a cycle in which they are both produced and consumed. We call such metabolites *self-regenerating* and we observe that they can be used to generate other metabolites that are not potential precursors. Self-regenerating metabolites and the metabolites they enable to be generated will be called the *continuously available* metabolites.

Let us consider the network of Figure 1 with A and C as potential precursors and E as target metabolite. In this case, B and D are self-regenerating metabolites that are *continuously available*. Note that G is also a *continuously available* metabolite.

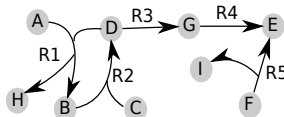


Fig. 1: A metabolic network  $G$  with set of metabolites  $\mathcal{C} = \{A, B, C, D, E, F, G, H, I\}$ , set of reactions  $\mathcal{R} = \{R1, R2, R3, R4, R5\}$

For each reaction  $r \in \mathcal{R}$ , we call  $Inp(r)$  the set of substrates of  $r$  and  $Out(r)$  the set of products of  $r$ . In what follows,  $\mathcal{P}(\mathcal{S})$  denotes the power set of a set  $\mathcal{S}$ .

**Definition 1.** Given  $X \in \mathcal{P}(\mathcal{C})$ ,  $Reach(X) \in \mathcal{P}(\mathcal{C})$  is the set of compounds  $y$  for which there exists  $r \in \mathcal{R}$  with  $Inp(r) \subseteq X$  and  $Out(r) \ni y$ .

In other words,  $y \in Reach(X)$  if there exists a reaction in  $\mathcal{R}$  producing  $y$  whose substrates are in  $X$ .

Given sets  $X$ , compounds in infinite supply, and  $Z$ , *continuously available* compounds, we wish to compute the total set of compounds that can be produced by the network.

**Definition 2. (Reachability function)** Let  $X \in \mathcal{P}(\mathcal{C})$  and  $Z \in \mathcal{P}(\mathcal{C})$  be two subsets of metabolites. The reachability function  $f_Z : \mathcal{P}(\mathcal{C}) \rightarrow \mathcal{P}(\mathcal{C})$  is defined as  $f_Z(X) = X \cup \text{Reach}(X \cup Z)$ .

We define the functions  $f_Z^k(X) = f_Z(f_Z^{k-1}(X))$ , with  $f_Z^1(X) = f_Z(X)$ , as the function obtained by iterating  $k$  times the function  $f_Z$ .

**Definition 3. (Scope function)** Let  $Z \in \mathcal{P}(\mathcal{C})$  and  $X \in \mathcal{P}(\mathcal{C})$  be two subsets of metabolites. The scope function  $f_Z^* : \mathcal{P}(\mathcal{C}) \rightarrow \mathcal{P}(\mathcal{C})$  is  $f_Z^*(X) = f_Z^k(X)$  for any  $k$  such that  $f_Z^k(X) = f_Z^{k+1}(X)$ .

Note that  $f_Z$  is monotone, both in  $X$  and in  $Z$  and  $f_Z^*$  represents what may be produced from  $X$  with the help of  $Z$  and using reactions in  $\mathcal{R}$ . To define when a set of compounds  $X$  is a precursor set of a target  $T$ , we therefore need to impose that  $f_Z^*$  contains  $T$ , and  $f_Z^*$  can "regenerate"  $Z$ .

**Definition 4. (Precursor set)** A set of metabolites  $X \subseteq \mathcal{P}(\mathcal{C})$  is a precursor set of  $T \subseteq \mathcal{P}(\mathcal{C})$  if there exists a set  $Z \subseteq \mathcal{P}(\mathcal{C})$  such that  $f_Z^*(X) \supseteq T \cup Z$ .

Note that in the above definition we are interested only in the existence of  $Z$  and not in characterising the set of available compounds that allows  $X$  to be a precursor set of  $T$ .

We now define a precursor set of a single target  $t$  in the hypergraph model.

**Definition 5. (Hyperpath with a set of continuously available metabolites)** A set  $H(X, Z, t) \in \mathcal{P}(\mathcal{R})$  of reactions is a hyperpath from a set of metabolites  $X$  to  $t$  using another set of metabolites  $Z$  if it satisfies:

1. The reactions in  $H(X, Z, t)$  can be ordered  $\langle r_1, r_2, \dots, r_k \rangle$  so that:
  - i) for all  $r_i$ ,  $\text{Inp}(r_i) \subset X \cup Z \cup \text{Out}(r_1) \cup \dots \cup \text{Out}(r_{i-1})$ ;
  - ii)  $t \in \text{Out}(r_k)$ ;
  - iii) for all  $s \in Z$ , there exists  $j(s)$  such that  $s \in \text{Out}(r_{j(s)})$ ,
2. No proper subset of  $H(X, Z, t)$  verifies the above.

Clearly, if there is a hyperpath  $H(X, Z, t)$  then  $X$  is a precursor set of  $t$ . For instance, in the Figure 1, if  $E$  is the target, and  $A$  and  $C$  are potential precursors, there is a hyperpath  $H(A, C, B, E) = r_1, r_2, r_3, r_4$  such that the set  $A, C$  is a precursor set of  $E$ .

The reverse is shown in the following.

**Lemma 6.** If  $X$  is a precursor set of  $t$ , then there exists a hyperpath  $H(X, Z, t)$  for some  $Z \in \mathcal{P}(\mathcal{C})$ .

**Sketch of the proof.**

In the following we define recursively a sequence of reactions: starting from the target  $t$ , at each step  $i$  we choose a reaction  $r_i$  that produces a non-precursor and/or a substrate not yet produced by some of the previous reactions  $r_1, \dots, r_{i-1}$ . The obtained sequence may contain repetitions. By eliminating repetitions and inverting the list, we get an ordered set  $H$  that fullfills condition 1 of Definition 5.

In order to find reactions that can be reached from  $X$ , we only consider reactions in the set  $W = \{r \in \mathcal{R} \mid \text{Inp}(r) \subseteq f_Z^*(X)\}$ , i.e. reactions that takes as substrates only compounds available in the scope of  $X$ .

Let  $N_0 = \{t\}$  and  $A_0 = \emptyset$ . At iteration  $i$ ,  $i \geq 1$ , we define the sets  $A_i = \cup_{j=1}^i \text{Out}(r_j) \setminus X$  and  $N_i = \cup_{j=1}^i \text{Inp}(r_j) \setminus (A_i \cup X)$ , i.e.,  $A_i$  is the set of compounds produced in the first  $i$  reactions and  $N_i$  the set of compounds consumed but not yet made available in the first  $i$  reactions. In iteration  $i$ , select some  $c_i \in N_{i-1}$ . Since  $c_i \notin X$ , we know, by definition of  $W$ , that there exists a reaction  $r_i \in W$  with  $c_i \in \text{Out}(r_i)$ . We update the set  $A_i$  by  $A_{i-1} \cup (\text{Out}(r_i) \setminus X)$  and define the set  $S_i = \text{Inp}(r_i) \cap A_i$ , substrates of  $r_i$  that have been produced already. We update  $N_i = (N_{i-1} \cup \text{Inp}(r_i)) \setminus (A_i \cup X)$ .

This process is iterated until  $N_i$  is empty, which has to occur since the sequence of  $A_i$  is monotone. Let  $k$  be the first (and last) iteration such that  $N_k = \emptyset$  and let  $Z = \cup_{i=1}^k S_i$ . The sequence  $\omega = r_1, \dots, r_k$  may have repetitions. We define  $\bar{r}_1, \dots, \bar{r}_\ell$  as the subsequence that contains only the first occurrence of each reaction and define  $H = \{\bar{r}_1, \dots, \bar{r}_\ell\}$  to be the set including all these reactions.

In the full version we shall show that the above defined set of reactions fullfills the conditions of Definition 5.  $\square$

In the following we study the three problems below:

**Problem MAL-PS( $G, P, T$ ):** given a metabolic network  $G = (\mathcal{C}, \mathcal{R})$  with  $P \subset \mathcal{C}$  the set of all potential precursors and  $T \subset \mathcal{C}$  the set of target metabolites, find a minimal precursor set  $X \subset P$  of  $T$  in  $G$ .

**Problem MIN-PS( $G, P, T$ ):** given a metabolic network  $G = (\mathcal{C}, \mathcal{R})$  with  $P \subset \mathcal{C}$  the set of all potential precursors and  $T \subset \mathcal{C}$  the set of target metabolites, find a minimum size precursor set  $X \subset P$  of  $T$  in  $G$ .

**Problem ALLMAL-PS( $G, P, T$ ):** given a metabolic network  $G = (\mathcal{C}, \mathcal{R})$  with  $P \subset \mathcal{C}$  the set of all potential precursors and  $T \subset \mathcal{C}$  the set of target metabolites, enumerate all minimal precursor sets  $X \subset P$  of  $T$  in  $G$ .

Given the network of Figure 1, let  $E$  be the target and  $\{A, C, F\}$  the potential precursors. A solution to the MAL-PS( $G, P, T$ ) problem is  $\{A, C\}$  or  $\{F\}$  whereas the precursor set  $\{A, C, F\}$  is not a solution because it is not minimal. The solution to the MIN-PS( $G, P, T$ ) problem is  $\{F\}$ . Finally, the solution to the ALLMAL-PS( $G, P, T$ ) problem is given by  $\{A, C\}$  and  $\{F\}$ .

### 3 Minimal and minimum precursor sets

The following useful Lemma is an interesting result in itself.

**Lemma 7.** *Given  $G, T$ , there exists a polynomial time algorithm to check whether a set  $X$  is a precursor set of  $T$  in  $G$ .*  $\square$

The following algorithm for  $\text{MAL-PS}(G, P, T)$  resembles the one presented by Handorf *et al.* [4]. Given  $G, P$  and  $T$ , a simple algorithm to solve  $\text{MAL-PS}(G, P, T)$  first sets  $X = P$ ; at the beginning all compounds in  $X$  are unmarked. Then, the algorithm determines whether  $X$  is a precursor set of  $T$  in  $G$  using the algorithm in the proof of Lemma 7. If the answer is negative, then there is no precursor set and the algorithm stops. Otherwise, let  $u$  be an arbitrary unmarked compound of  $X$  and set  $X' = X - \{u\}$ ; the algorithm determines whether  $X'$  is a precursor set of  $T$  in  $G$ . If so, then  $u$  is deleted from  $X$ : there exists at least one minimal solution that does not contain  $u$ . If not, then  $u$  remains in  $X$  and it is marked. The algorithm iterates this procedure until no unmarked compounds are left. Since all marked compounds are essential for a precursor set, they form a minimal precursor set of  $T$  in  $G$ . The following theorem states the correctness of the algorithm.

**Theorem 8.** *Given  $G, P, T$ , there exists a polynomial time algorithm that solves  $\text{MAL-PS}(G, P, T)$ .* □

The following is proved by a reduction from the Hitting Set problem [3].

**Theorem 9.**  *$\text{MIN-PS}(G, P, T)$  is NP-hard.* □

## 4 Algorithm for enumerating all minimal precursor sets

To facilitate the exposition, we consider the case of a single target metabolite. The solution for several target metabolites is computed by adding an artificial node to the metabolic network and one irreversible reaction that has each target as substrate and the artificial node as only product, which then acts as the single target.

The algorithm is composed of two steps: the first one defines a special structure, called a *replacement tree*, that contains a representation of at least one hyperpath (see Section 2) for each precursor set of  $t$ . To achieve this, we proceed in a way that is analogous to the one adopted in the proof of Lemma 6; the main difference is that in this case  $X$  is unknown (in fact the algorithm is seeking all  $X$  that are precursor sets for  $t$ ). Therefore, when the algorithm moves backwards in the network starting from  $t$ , it must consider all reactions and not only those in  $W$ . At the end of step 1, the replacement tree will contain a representation of at least one hyperpath for each minimal precursor set of  $t$  but also the representation of some hyperpaths that do not represent minimal precursor sets. In the second step, the replacement tree is used to enumerate all the precursor sets for the target metabolite, and sets of metabolites that are not precursor sets are removed.

The proof of the correctness of the algorithm uses Lemma 6 and is omitted. The time and space complexity of the algorithm are linear in the size of the replacement tree whose size can be exponential (note also that the number of solutions might in any case be exponential in the size of  $P$ ). We finally observe

that the two steps are described separately for the sake of clarity: in fact they can be executed simultaneously, thereby improving the time and space requirements.

### Building the replacement tree

The replacement tree is rooted and directed from its root to the leaves. Nodes of the tree are labelled either by a metabolite or by a reaction. In the first case, we have a *metabolite node* while in the second a *reaction node*. The children of a metabolite node are reaction nodes labelled by those reactions that produce the metabolite while the children of a reaction node are labelled by its substrates. In the tree, both metabolite and reaction nodes have only one parent whereas both can have several children. In the sequel, we use the terms *product node* for the parent of a reaction node and *substrate nodes* for the children of a reaction node.

The construction of the tree starts at the root  $t$ . For each reaction producing this metabolite, we create a reaction node, which has as parent the root, and as children new metabolite nodes corresponding to the substrates of the considered reaction. In this way, we obtain a tree (of depth 3) whose leaves are metabolites. This process is then iterated for each new metabolite node.

Let us call  $u$  a newly created metabolite node and  $c$  the corresponding metabolite of  $u$ . Along any branch of the tree, the process stops when one of these three conditions below is verified:

1.  $c$  corresponds also to an ancestor of  $u$ ,
2.  $c$  corresponds to a child of one reaction node ancestor of  $u$ , or
3.  $c$  is not produced by any reaction (we cannot go further back in the network).

In the first case, we flag  $u$  as “continuously available”. An example is metabolite node  $u_1 = \text{H}$  that is one of the children of N12 in the tree depicted in Figure 2. In this case, this metabolite node is flagged as continuously available (indicated by a star beside the node). Indeed, the metabolite is regenerated by the network.

In the second case,  $c$  is considered as a child of a reaction node of  $u$ ; so it is not necessary to duplicate the search for a precursor by expanding the metabolite. An example is metabolite  $u_2 = \text{l}$  in Figure 2. In this case,  $\text{l}$  is produced by reaction node N14 which is not ancestor of  $u_2 = \text{l}$  in the tree. Since  $\text{l}$  is the child of the reaction node N13 ancestor of  $u_2 = \text{l}$ ,  $\text{l}$  has been already analysed. In the third case,  $c$  cannot be produced by any other reaction node so there is no need to expand it.

Therefore, when the process stops, all the leaves of the tree contain only metabolites not produced by any reaction or already visited metabolites.

The procedure implies that, for any solution  $X$  for target  $t$ , there exists a subtree whose set of reactions represents a hyperpath from  $X$  to  $t$  using some compounds that are regenerated (set  $Z$  in Lemma 6).

**Lemma 10.** *If  $X$  is a precursor set of  $t$ , then there exists a subtree of the replacement tree containing a hyperpath  $H(X, Z, t)$  for some  $Z \in \mathcal{P}(C)$ .*

### Enumerating the solutions

We now use the replacement tree to enumerate all minimal precursor sets for  $t$ .

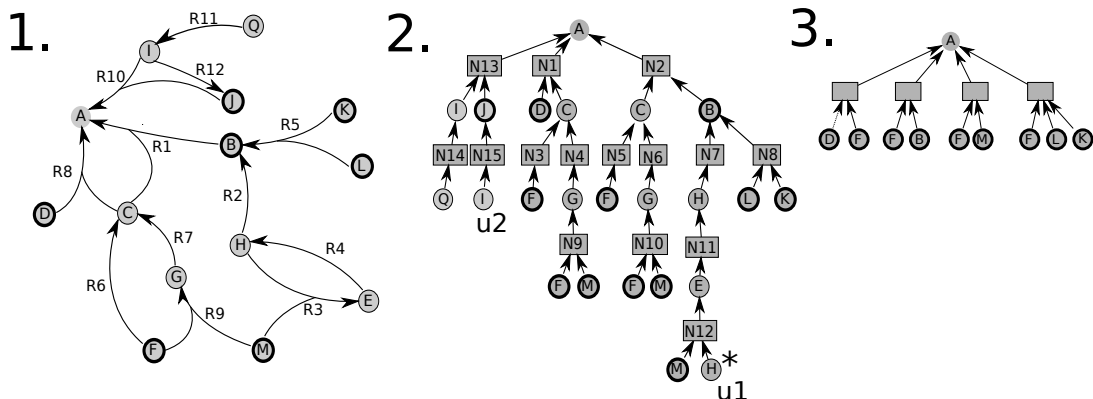


Fig. 2: An example of a metabolic network and of a replacement tree for the target metabolite A. **1.** The metabolic graph. The metabolite nodes surrounded in black are potential precursors. These are B, D, F, L, J, K and M. **2.** The replacement tree before compression. **3.** The replacement tree after compression. Each set of substrate nodes of a reaction node corresponds to a minimal precursor set.

This is done by successively processing subtrees that have a single reaction node  $r$  as root and one or more metabolite nodes as children that are all leaves in the tree. Depending on what those metabolite leaf nodes are (potential precursors, flagged metabolites or non-flagged metabolites), the subtree will either be eliminated, or it will be used to create a new subtree that will replace it. This effects a progressive compression of the original replacement tree until it has only three levels composed of the root (level one), the reaction nodes producing the root (level two) and the minimal precursor sets (level three). The final compressed tree preserves the same properties as the initial tree with respect to the minimal precursor sets that produce the target  $t$ .

The compression algorithm starts by considering a reaction node whose substrate nodes are all leaves. Let  $r$  be the label of such a node,  $p$  be the parent of  $r$ ,  $S$  the set of labels of its children and  $n$  the parent of  $p$  (note that  $S$  is a set of compounds and  $n$  is a reaction node). If  $r$  is such that either *i*) at least one of its substrate nodes is neither a potential precursor nor a flagged (*continuously available*) metabolite or *ii*) the potential precursors in  $S$  form a superset of the potential precursors that are substrate nodes of another reaction node having also  $p$  as parent, then the subtree rooted at  $r$  is simply eliminated from the tree.

If  $r$  is not eliminated, the subtree rooted at  $n$  is duplicated; namely all reaction and metabolite nodes are duplicated maintaining their label. Let  $n'$  be the root of this new subtree,  $p'$  the child of  $n'$  that corresponds to  $p$ , and  $r'$  the child of  $p'$  that corresponds to  $r$ . Furthermore,  $n'$  has the same parent as  $n$ .

We now modify the subtrees rooted at  $n$  and  $n'$  as follows:

- the metabolite nodes that are children of  $r$  in the subtree rooted at  $n$  are disconnected from  $r$  and  $r$  itself is eliminated;

- we replace node  $p'$  in the subtree rooted at  $n'$  with the set of children of  $r'$ . Both  $p'$  and  $r'$  are removed from  $n'$ .

We describe one example, using the replacement tree of Figure 2. Assume that, at some iteration of the compression algorithm, the subtree rooted at the reaction node N9 whose substrate nodes are leaves of the tree is considered. The children of N9 are potential precursors, and the parent of N9 has no other child. Therefore we cannot eliminate N9. Since the reaction node that is its immediate ancestor is N4 the subtree rooted at N4 is duplicated. Suppose the root of the duplicate subtree is labelled N4'. N4' is made the child of the parent of N4 labelled C. The metabolite nodes labelled F and M children of N9 are made the children of N4' in replacement of the copy of N4's only child labelled G. Reaction node N9 is removed. The parent of N4 has now 3 children: N3, N4 and N4'. If N4 is the new subtree considered at the next iteration of the algorithm, the algorithm would eliminate it since its only child is not a flagged leaf. If the subtree rooted at N4' is considered then its two children, labelled F and M, are potential precursors. However, since  $\{F, M\}$  is a superset of the set of potential precursors that are substrate nodes of N3 (only F), the subtree rooted at N4' can be eliminated and the next reaction node considered could be N3.

The process above described continues until the final compressed tree has only 3 levels: the root labelled by the target, the reaction nodes produced by the compression and the substrate nodes of these reaction nodes. The crucial property is that each step of the compression does not eliminate any minimal precursor set of  $t$ ; it follows that labels of the children of a level 2 node directly correspond to a minimal precursor set of the target, as stated in the following lemma.

**Lemma 11.** *If  $X$  is a minimal precursor set of  $t$ , then there exists a child  $x$  of the root of the final compressed tree such the set of labels of the children of  $x$  coincides with  $X$ .*  $\square$

## 5 Illustrations

We now briefly present two illustrations of our method. In the two cases, the set of potential precursors are defined in the same way. Since precursors are in general expected to be at the periphery of the network, we define as potential precursors all the metabolites not produced by any reaction and those that are involved in only one reaction which is reversible. This definition may not be sufficient for defining all potential precursors. We can therefore add some internal metabolites as further potential precursors. The final set and the procedure to get the metabolic data are given in the supplementary file available at <http://biomserv.univ-lyon1.fr/cottret/WABI/annexes.pdf>.

### 5.1 Checking inconsistencies in a metabolic database

The first example is very similar to the study done by Romero and Karp [8] whose goal was to check inconsistencies in the metabolic database ECOCYC [6].

There are two steps: the first proceeds exactly as in [8] on ECOCYC and is done to recover the target metabolites not reached by Romero and Karp’s forward propagation algorithm. From the set of nutrients and of bootstrap metabolites given as input (these compounds are indicated by the user as being always present even though they may not be *continuously available* metabolites), the target metabolites not present in the scope of the nutrients are identified. In the second step, our method is applied on the network not produced by the forward propagation to find all the minimal precursor sets of the target metabolites not reached during the first step.

**Data.** The metabolic network we build contains 897 metabolites and 879 reactions, of which 104 are defined as irreversible.

The metabolites that are defined as bootstrapping during the first step, (which can be used during the forward propagation to fire a reaction), are those present in the minimal growth medium as indicated in [8], plus the first ten most connected metabolites and some metabolites whose presence in the cell seem obvious such as Coenzyme-A (see the supplementary file for a table of these bootstrapping metabolites). The only input metabolite for the forward propagation is glucose. The target metabolites are the 20 amino acids.

**Results.** In the first step, the forward propagation method in [8] returns a network with 513 reactions and 437 metabolites. This means that half of the network can be directly produced by injection of glucose, taking into account the presence of the bootstrapping metabolites selected in [8]. This subnetwork would even be bigger if our *continuously available* metabolites were also considered during the forward propagation. Among the 20 amino acids which are the building blocks for the synthesis of proteins, only 2 are not produced by the forward propagation: lysine and methionine.

Applying our method with lysine as target metabolite returns 9 sets of potentially missing precursors. Among them, one was noticeable: a set which contains only tetrahydrodipicolinate. Indeed, this metabolite is involved in the biosynthesis of lysine and, surprisingly, the pathway appears complete in *Escherichia coli*. Therefore the metabolite should not be a precursor. In fact, a closer look at the data reveals an error in ECOCYC. Indeed, tetrahydrodipicolinate is a substrate of reaction 1.3.1.26 which is indicated as irreversible, and furthermore in the “wrong” direction relatively to the known pathway. Once this reaction is made reversible, both lysine and methionine become present in the sub-network produced by the forward propagation.

## 5.2 Finding nutrients necessary to a metabolic function

The second example tests a case when the number of potential precursors is expected to be high. This is the case of the metabolic network of the endosymbiotic bacterium *Carsonella ruddii*. Indeed, this bacterium lives inside specialised cells of psyllids, phloem sap-feeding insects. *Carsonella ruddii* has the most reduced among known metabolic networks [7], hence the expected high number of potential precursors for its essential metabolites, such as the amino acids

the bacterium provides to its host, who is not capable of producing them. Yet recently, the analysis of the bacterium genome [9] showed that half of the pathways involved in the biosynthesis of essential amino acids have been completely or partially lost. The bacterium therefore requires possibly many nutrients from the host to enable it to fill in these “holes”. As an illustration, we chose to search for the precursors of one such essential amino acid, the arginine whose metabolic pathway appears to be complete in the bacterium.

**Data.** We built a metabolic network containing 130 compounds and 71 reactions, 16 of which are defined as irreversible. As in the previous example, the bootstrapping compounds defined in [8] were eliminated from the network.

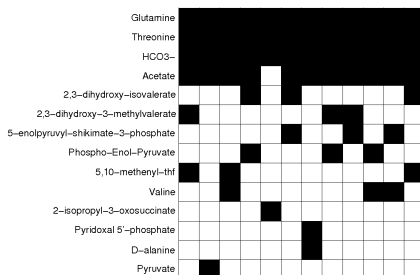


Fig. 3: The sets of precursors of the arginine in *Carsonella ruddii*

**Results.** We found 12 minimal precursor sets for arginine. The results are presented in Figure 3. One interesting thing in these results is that glutamine, threonine, and ion bicarbonate are present in all solutions. Glutamine and ion bicarbonate are involved in reaction 6.3.5.5, which represents an essential step in the arginine biosynthesis pathway, as described in METACYC [1]. Threonine as precursor of arginine deserves to be discussed. The path between threonine and arginine goes through two different metabolic pathways by reversible reactions to produce aspartate, a key metabolite of the arginine biosynthesis. Interestingly, those two metabolic pathways (the “threonine biosynthesis from homoserine”, and the “homoserine biosynthesis” pathways) are traversed in a direction inverse to the one classically indicated in the database for those specific pathways. Of course, this may be an artefact due to an imprecision concerning the direction of the reactions, but it may also mean that those reactions can be used in a direction inverse to the one indicated in the reference metabolic pathways.

## 6 Conclusion and perspectives

We proposed the first topology-based exact method to find minimal precursor sets for a set of target metabolites. Despite the complexity of the problem, the method can be applied to genome-scale metabolic networks.

In contrast with previous methods, we deal in a formally clear way with the issue of cycles when searching for precursors in a given metabolic network.

To this purpose, we defined the notion of “continuously available” compounds, which either are able to self-regenerate themselves once activated, or to be generated with the aid of self-regenerating metabolites. An implementation of the enumeration algorithm appears to allow finding all minimal precursor sets for networks of the sizes used in the examples in a time ranging from a few seconds to a few hours.

Our analyses show that some concepts would need further refinements. For instance, the assumption that all potential precursors are always in infinite supply from the environment may not be fully realistic from a biological point of view. Indeed, some nutrients are always available in some environmental conditions and not available in other conditions. The possibility to define different sets of potential precursors allows to search for the precursors in certain environments but needs some *a priori* biological information about the nutrients available in each condition.

Compressing the replacement tree while building it enables to reduce space while enumerating all solutions. Efficient though this is, we believe better results might be achievable. Both this and the previous problem require to define and deal well with cycles in hypergraphs representing metabolic networks.

**Acknowledgements** This work was funded by the ANR (Reglis project NT05-3\_45205), the ANR and the BBSRC (MetNet4SysBio project ANR-07-BSYS 003 02), the INRIA, the CNRS and the French Ministry of Foreign and European Affairs (STIC-AmSud project) and the Dutch BSIK-BRICKS project. The authors would like also to thank the bioinformatic team at the Genoscope who provided us with the metabolic network data.

## References

1. R. Caspi et al. MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res*, 34(Database issue):D511–D516, 2006.
2. C. Francke, R. J. Siezen, and B. Teusink. Reconstructing the metabolic network of a bacterium from its genome. *Trends Microbiol*, 13(11):550–558, Nov 2005.
3. M. R. Garey and D. S. Johnson. *Computers and Intractability (A guide to the theory of NP-completeness)*. W.H. Freeman and Company, New York, 1979.
4. T. Handorf, N. Christian, O. Ebenhöf, and D. Kahn. An environmental perspective on metabolism. *J Theor Biol*, Nov 2007.
5. M. Kanehisa et al. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*, 34(Database issue):D354–D357, 2006.
6. I. M. Keseler et al. EcoCyc: a comprehensive database resource for Escherichia coli. *Nucleic Acids Res*, 33(Database issue):D334–D337, 2005.
7. A. Nakabachi et al. The 160-kilobase genome of the bacterial endosymbiont *carsonella*. *Science*, 314:267, 2006.
8. P. R. Romero and P. Karp. Nutrient-related analysis of pathway/genome databases. *Pac Symp Biocomput*, pages 471–482, 2001.
9. J. Tamames, R. Gil, A. Latorre, Peretó, F. Silva, and A. Moya. The frontier between cell and organelle: genome analysis of candidatus *carsonella ruddii*. *BMC Evol Biol*, 7:181, 2007.