MINIMUM MUTATION ALGORITHM FOR GAPLESS METABOLIC NETWORK EVOLUTION

Esa Pitkänen, Juho Rousu
Department of Computer Science, University of Helsinki, P.O. Box 68, FI-00014 Helsingin yliopisto, Finland

Mikko Arvas
VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Espoo, Finland

Keywords: Algorithms, Bioinformatics, Evolution, Fungi, Metabolism, Parsimony, Phylogeny, Systems Biology.

Abstract: We present a method for inferring the structure of ancestral metabolic networks directly from the networks of observed species and their phylogenetic tree. Our method aims to minimize the number of mutations on the phylogenetic tree, whilst keeping the ancestral networks structurally feasible, i.e., free of reaction gaps. To this end, we present a parsimony-based method that generates metabolic network phylogenies where the ancestral nodes are required to represent gapless metabolic networks, networks where all reactions are reachable from external substrates. In particular, we introduce the gapless minimum mutation problem: finding phylogenies of gapless metabolic networks when the topology of the phylogenetic tree is given, but the content of ancestral nodes is unknown. The gapless minimum mutation problem is shown to be computationally hard to solve even approximatively. We then propose an efficient dynamic programming based heuristic that combines knowledge on both the metabolic network topology and phylogeny of species. Specifically, the reconstruction of each ancestral network is guided by the heuristic to minimize the total phylogeny cost. We experiment by reconstructing phylogenies generated under a simple random model and derived from KEGG for a number of fungal species.

1 INTRODUCTION

Modelling of metabolism is essential in a variety of applications of biotechnology and medicine including bioprocess development (Raman and Chandra, 2009), study of metabolic diseases (Sigurdsson et al., 2009) and drug target identification (Jamshidi and Palsson, 2007). Global characteristics of cellular metabolism by metabolic networks have been studied intensively by a variety of computational approaches, including metabolic reconstruction (see (Pitkänen et al., 2010) for a recent survey), metabolic flux analysis (Palsson, 2006), 13C isotopic tracing (Rantanen et al., 2008) and structural analysis of metabolic networks (Lacroix et al., 2008).

The structure of the metabolic network is a major contributor to the phenotypes that an organism manifests. Metabolic networks have been shown to be scale free (i.e., the networks contain hub metabolites of high connectivity) and modular (Kreimer et al., 2008). The structure is known to constrain the phenotypes the network can realize (Palsson, 2006), thus the structure is also likely to be conserved in evolution (Wagner, 2009).

Recently, the increasing number of fully sequenced genomes has enabled comparative genomics analysis of metabolic network evolution (for review see (Caetano-Anollés et al., 2009)). Many computational approaches have concentrated on deriving rigorous measures of biological network and pathway similarity (Sharan and Ideker, 2006), thus enabling construction of phylogenies of networks with distance-based methods. Particularly methods for metabolic network and pathway comparison have been developed (Dandekar et al., 1999; Tohsato et al., 2000; Clemente et al., 2007; Mano et al., 2010).

Distance-based methods do not immediately yield predictions on the contents of ancestral networks, however. Knowledge on ancestral networks is important as it may shed light on the evolutionary mechanisms that have generated the observed networks. An approach complementary to distance-based methods...
often used to give insight on ancestral node contents in phylogenetic trees is maximum parsimony, where one tries to find ancestral objects which minimize the total number of evolutionary changes required to explain the observed data. The maximum parsimony principle has been utilized in many domains, for instance in the analysis of sequence data (Fitch, 1971; Sankoff, 1975; Clemente et al., 2009; Tuller et al., 2010) as well as gene regulatory networks (Bourque and Sankoff, 2004).

A direct application of maximum parsimony methods to biological network data generally results in structurally infeasible ancestral networks. For instance, consider the two metabolic networks $Y$ and $Z$ with a common immediate ancestor $X$ shown in Figure 1. A parsimonious scenario (top right) includes an ancestral network where a metabolite is required by the network but cannot be produced, suggesting that the pathway $m \rightarrow c$ cannot operate and the network is infeasible.

Graph evolution models taking into account dependencies imposed by the network structure have been proposed. For instance, Mithani et al. gave a Markov process for simulating metabolic network evolution under a neighbor dependency model where appearance of a reaction depends on the fraction of neighboring reactions already present in the network (Mithani et al., 2009; Mithani et al., 2010). However, they reported results only for relatively small metabolic networks.

In this paper, we introduce a computational method for reconstructing ancestral metabolic networks in a given phylogenetic tree. The method combines the maximum parsimony principle with the requirement that the resulting networks are plausible in terms of network connectivity, thus contributing towards bridging the gap between the structural and phylogenetic analysis of metabolic networks. Specifically, our method builds phylogenies of metabolic networks where the ancestral nodes of the phylogeny adhere to structural network constraints: The networks are required to be free of reaction gaps, that is, reactions whose substrates cannot be produced from external metabolites. The choice of external metabolites reflects the estimated metabolic environment: the organisms are assumed to have them available in abundance and possess the necessary transports. To this purpose, computational methods have been developed to identify a set of minimal nutrients, given metabolic network structure (Handorf et al., 2008; Borenstein et al., 2008).

In section 2, we formulate the gapless minimum mutation problem where the topology of the phylogenetic tree is taken as input and the problem is to infer the structure of the ancestral networks so that the total phylogeny cost is minimized. We show the problem to be computationally hard to even approximate and go on to propose an efficient heuristic algorithm, which solves the problem well in practise. In section 3, we experiment with the algorithm in two scenarios: First, we analyze randomly generated and perturbed data. Second, we study gapless phylogeny reconstruction for a collection of fungal species. Section 4 ends the paper with conclusions.

2 METHODS

We are interested in metabolic networks that are functional in the sense that the network is able to produce substrates of all its reactions from some given set of source metabolites. Such networks are termed gapless, with a precise definition given below.

A metabolic network can be described as a binary string $N \in \{0,1\}^m$, where each $N_i = 1$ states that the reaction $r_i$, drawn from a collection of reactions $R$, is in the network. We use the shorthand $r_i \in N$ when $N_i = 1$. Further we assume a set of metabolites $M$ is consumed and produced by the $m$ reactions. The set of substrate and product metabolites of a reaction $r_i$ are given by $S(r_i) \subset M$ and $P(r_i) \subset M$, respectively.

To see how a string $N$ encodes metabolic network connectivity, note that $N$ induces a directed bipartite graph $G(N) = (V,E)$, with a node $v_r \in V$ for each reaction $r \in N$. Additionally each metabolite $b \in S(r_i) \cup P(r_i)$ for every $r_i \in N$ contributes a node $v_b \in V$. Edges $(v_r,v_b) \in E$ and $(v_b,v_r) \in E$ are added whenever reaction $r$ produces or consumes metabolite $b$, respectively. Figure 2 shows this graph representation implicitly encoded by a string $N$ for five reactions.

We first define gaplessness in terms of reactions that are reachable from a set of source metabolites $S$ (Pitkänen et al., 2005).

**Definition 2.1.** Let $N$ be a metabolic network and $S \subseteq M$ be a set of source metabolites.

- A reaction $r \in N$ is reachable from $S$ in $N$ if all its substrates $S(r)$ are reachable from $S$ in $N$.
- A metabolite $b \in M$ is reachable from $S$ in $N$ if either $b \in S$ or $b \in P(r)$ for some reaction $r \in N$ that is reachable from $S$ in $N$.

A gapless metabolic network $N$ under $S$ is a metabolic network where all reactions are reachable from $S$ in $N$.

We often omit an explicit mention of the source set $S$ if it is clear in the context, saying only that a metabolic network is gapless. If a reaction $r \in N$ is
not reachable from \( S \), we say that \( r \) is a reaction gap (under \( S \)). In addition, we say that a metabolic network is gapped, if it contains at least one reaction gap. Figure 2 illustrates these concepts.

### 2.1 Gapless Minimum Mutation Problem

We next introduce the computational problem of finding a gapless phylogeny when the tree topology and input taxa are given.

**Problem 2.2. Gapless Minimum Mutation problem (GMM).** Given a reaction collection \( \mathcal{R} \), \( m = |\mathcal{R}| \), a rooted binary tree \( T = (V,E) \), labeling \( L(u) \in \{0,1\}^m \) specifying a metabolic network for each leaf node and source metabolites \( S \), find a labeling for each internal node of \( T \) such that

1. \( c = \sum_{(u,v) \in E} d(L(u), L(v)) \) is minimized and
2. \( L(u) \) is a gapless metabolic network under \( S \) for each internal node \( u \in V \),

where \( d \) is Hamming distance.

The equivalent problem defined for binary strings without the gapless constraint (2), Minimum Mutation problem, can be solved in polynomial time with the Fitch algorithm because each character position can be solved independently of each other (Fitch, 1971; Gusfield, 1997). However, in contrast to the Minimum Mutation problem, the character positions in GMM are not necessarily independent of each other: setting a certain \( L_i(u) \in \{0,1\} \) may impose constraints on other positions \( j \neq i \) due to the gapless constraint. Note that the taxa contained in the leaves may or may not correspond to gapless metabolic networks.

**Theorem 2.3.** Deciding whether a solution with cost \( c \leq k \) to Gapless Minimum Mutation problem exists given \( k \in \mathbb{N} \) is NP-complete.

**Proof.** Given a solution to GMM, we can both compute the cost \( c \) and check that each network \( L(u) \) is gapless in polynomial time (Pitkanen et al., 2005), hence the problem is in NP.

To show that the problem is NP-hard, we reduce the well-known NP-complete Minimum Set Cover problem (Garey and Johnson, 1979) to Gapless Minimum Mutation problem. Let \( X \) be a finite set and \( C \) be a collection of subsets of the set \( X \). The Minimum Set Cover problem, we ask for the smallest sub-

\[ C' \subseteq C \]

such that every element of \( X \) belongs to at least one member of \( C' \). To create an instance of GMM, we first set up a reaction collection \( \mathcal{R} \) with two groups of reactions, one group for items in \( X \), another for sets in \( C \). Specifically, let \( \mathcal{R} \) contain a reaction \( r_i \) for each \( x_i \in X \) with \( S(r_i) = \{ b_i \} \) and \( P(r_i) = \{ c_i \} \), and a reaction \( q_j \) for each \( C_j \in C \) with \( S(q_j) = \{ a_j \} \) and \( P(q_j) = \{ b_i | x_i \in C_j \} \). In addition, let \( \mathcal{R} \) contain a reaction \( x \) with \( S(x) = \{ c_i | x_i \in X \} \) and \( P(x) = \{ m \} \).

To set up a phylogenetic tree, let \( T = (V,E) \) be a binary rooted tree with \( V = \{ v_1, v_2, v_3 \} \), \( E = \{ (v_3, v_1), (v_3, v_2) \} \) and root \( v_3 \). Finally, let input metabolic networks at leaves be \( L(v_1) = L(v_2) = \{ r_i | x_i \in X \} \cup \{ x \} \) and the set of source metabolites...
Figure 2: Example metabolic network with reactions $N = \{r_1, \ldots, r_{10}\}$ and metabolites $\{m_1, \ldots, m_{10}\}$. When $S = \{m_1, m_2\}$ (double circles), reactions $r_4$ and $r_5$ (dotted rectangles) are reaction gaps in $N$. However, the network $N$ under $S' = \{m_1, m_2, m_3\}$ would be gapless, because then reaction $r_4$ would be reachable. On the other hand, $N' = \{r_1, r_2, r_3\}$ is gapless under $S = \{m_1, m_2\}$.

Figure 3 shows the reduction from an example set cover instance $X$ to make $L(v_3)$ gapless. Existence of $x$ will ensure that $L(v_3) \neq \emptyset$ when $|C| = |X|$. To transform a solution to GMM back to a solution to Minimum Set Cover, let set $C'$ contain $c_i$ for each $q_i$ assigned to node $v_3$. The reduction can be done in polynomial time. In reduction, each reaction $q_i$ is assigned to $v_3$ if and only if $c_i$ appears in the optimal solution, assuming without loss of generality a unique set cover solution. If there is no solution to the set cover instance, also GMM is unsolvable as it is not possible to fix $L(v_3)$ to be gapless. Thus, GMM reduces to the set cover problem, thus contradicting the assumption.

Often we have gapped metabolic networks as taxa to begin with. For instance, initial networks may be a result of function assignment by annotation transfer, where the resulting structure of the draft metabolic network is not of concern. However, these networks should also be functional and therefore we can attempt to fix them gapless while finding ancestral nodes. To do this, we can extend the tree $T$ such that for each leaf $u$ we add an edge $(u, u')$ and assign $L(u') = L(u)$. Solving GMM in the modified tree thus finds a solution where nodes $u'$ retain the original input networks but gaps in internal nodes $u$ are fixed. We provide an example of such situation in experiments, where we utilize gapped networks derived from a metabolic database.

### 2.2 Local Adjustment Algorithm

To overcome the computational complexity, we next propose an algorithm that solves the Gapless Minimum Mutation problem in two phases. In the first phase, the assignments corresponding to the minimum mutation cost are computed with the Fitch algorithm (Fitch, 1971; Gusfield, 1997). In the second phase, the tree is traversed top-down and a gapless metabolic network is assigned at each node by filling the gaps remaining after the Fitch pass. The algorithm relies on estimates on how much filling each gap would increase the total phylogeny cost, and attempts to choose gap-filling reactions which increase the cost as little as possible. The cost increase estimates are computed by the algorithm for each ancestral network.
In Algorithm 1 at line 2, we first compute equality sets $F_i(v)$ for each internal node $v$ and position $i$ with the Fitch algorithm. In a binary tree, equality set $F_i(v)$ is defined for each position $i$ as

$$F_i(v) = \begin{cases} \{L_i(v)\} & \text{if } v \text{ is leaf} \\ F_i(x) \cup F_i(y) & \text{if } v \text{ is not leaf and } F_i(x) \cap F_i(y) \neq \emptyset \\ F_i(x) \cup F_i(y) & \text{if } v \text{ is not leaf and } F_i(x) \cap F_i(y) = \emptyset \end{cases}$$

where $x$ and $y$ are the children of an internal node $v$ (Gusfield, 1997).

The tree is traversed in a top-down pass and an initial labeling $L(v)$ is decided according to the Fitch top-down phase (Algorithm 1, lines 3–12). Each initial labeling is then adjusted so that it satisfies the gaplessness constraint.

For each internal node $v$, the algorithm calls the subroutine MinFill, which returns a gapless network containing $L(v)$ and gapfill reactions (Algorithm 2). Particularly, if $L(v)$ is already gapless, it is returned as such. MinFill attempts to satisfy the gaplessness criterion by backtracking from each reaction not reachable from sources $S$ in network $L(v)$ and iteratively adding reactions to the fill set $\Gamma$. Subprocedure terminates when either all reactions have been reached or the algorithm notices that all gaps cannot be filled.

A heuristic is used to guide the backtracking phase by considering reaction assignments in the parent and child nodes. In particular, the algorithm attempts to choose reactions to the fill set $\Gamma$ such that the parsimony cost increase is minimized. To do this, we first compute distances $d_f$ that provide an estimate for each reaction how much the parsimony cost will increase compared to the optimal Minimum Mutation solution, if the reaction is added to the network. Formally, the distance $d_f$ is a lower bound to the increase in parsimony cost that is the result of adding reaction $r_i$ and other reactions required to make $r_i$ reachable to $v$:

$$d_f(v, r_i) = \max_{m: S(r_i) \subseteq Pr(m)} \min_{q \in Pr(m)} d_f(v, q) + d_y(v, r_i)$$

where $S(r_i)$ are the substrates of reaction $r_i$ and $Pr(m)$ are the reactions producing $m$. To introduce some notation, let $G_i(w, v) = \{L_i(w)\}$ iff $pa(v) = w$ and
Local adjustment algorithm returns \( d \) only and is defined as \( d_G(v, r_i) = \delta(v, \text{pa}(v), r_i) + \sum_{c \in \text{ch}(v)} \delta(v, c, r_i) \) with

\[
\delta(v, w, r_i) = \left\{ \begin{array}{ll}
1 & \text{if } G_i(w, v) = \{0\} \text{ and } L_i(v) = 0 \\
-1 & \text{if } G_i(w, v) = \{1\} \text{ and } L_i(v) = 0 \\
0 & \text{if } G_i(w, v) = \{0, 1\} \text{ and } L_i(v) = 0 \\
0 & \text{if } L_i(v) = 1
\end{array} \right.
\]

Note that at root \( v \), we set \( \delta(v, \text{pa}(v)) = 0 \).

Parsimony cost increase \( d_q \) at each reaction is always non-negative. To see this, we can enumerate the values for \( d_q \) at an internal node given \( G_i \) at parent and children (Table 1). Symmetric cases are omitted from the table, where a boldface \( 0 \) signifies that this combination would yield \( L_i(v) = 1 \) in the bottom-up phase of the algorithm and thus the reaction would already be in the network.

Subroutine MinDist is called from Algorithm 1 to compute distances \( d_f \) for all \( r \) with dynamic programming (Algorithm 3). If for some required substrate there are only producers \( r_i \) that have \( d_f(v, r_i) = \infty \), the algorithm fails as the required substrate cannot be reached from \( S \). In such cases, source set \( S \) needs to be expanded or reaction collection \( \mathcal{R} \) revised. To avoid loops, \( \varepsilon > 0 \) is added to distances \( d_r \), ensuring that distances strictly increase when traversing the network.

Algorithm 2: MinFill

1. Input: \( v, S, D, M \)
2. \( \Gamma \leftarrow \emptyset \), \( Q \leftarrow \{ r \in v \mid D(r) = \infty \} \)
3. while \( |Q| > 0 \) do
4. \( r \leftarrow \text{pop}(Q) \)
5. for all \( o \in S(r) \) do
6. if \( o \notin M \) then
7. \( q \leftarrow \text{argmin}_{q \in P(o)} D(q) \)
8. if \( D(q) = \infty \) then
9. return “Impossible to find gapfilling set”
10. if \( q \notin \Gamma \) and \( q \notin v \) then
11. \( \text{push}(Q, q) \)
12. \( \Gamma \leftarrow \Gamma \cup \{ r \} \)
13. Return \( \Gamma \)

Theorem 2.5. Local adjustment algorithm returns gapless metabolic networks \( L(u) \) for each internal node \( u \) or reports failure in \( O(nm \log m) \) time, where \( n \) is the number of species and \( m = |\mathcal{R}| \) is the number of reactions.

Proof. As each reaction is inserted at most once to the heap, Algorithm 3 takes \( O(m \log m) \) time to compute distances \( d_f \) assuming that both heap operations insert and extract-min take \( O(\log m) \) time. Further we assume that \( |P(r)| \) is bound by a constant, which is reasonable as typically in enzymatic reactions \( |P(r)| = 1, \ldots, 3 \). In Algorithm 2 each reaction is inserted at most once to the queue \( Q \), hence the subroutine takes \( O(m) \) time. Algorithm 1 makes a single call to subroutine Fitch which takes \( O(nm) \) time. Loop at line 11 is executed also \( O(nm) \) times. The time complexity of the algorithm is dominated by the \( O(n) \) calls to Algorithm 2, resulting in total \( O(nm \log m) \) time. Figure 4 illustrates the operation of the algorithm.

Note that it is possible to have an instance of GMM where the algorithm fails to find a gapless solution although such solution exists. As an example, consider two networks with a single reaction \( L(v_1) = L(v_2) = \{ r \} \), where \( S(r) = \{ m_1 \} \), \( P(r) = \{ m_2 \} \) and \( S = \emptyset \), and the tree of Figure 3. Then, the optimal solution is \( L(v_3) = \emptyset \). However, as the algorithm does not attempt to remove reactions from the initial Fitch solution \( L(v_3) = \{ r \} \), the network remains gapless. Such cases are avoided by carefully selecting the reaction collection and source metabolites. In practice, one must ensure that the source metabolite set is large enough for each reaction to be reachable in network \( L(u) = \mathcal{R} \).

3 EXPERIMENTS

We experimented with two datasets. First, we generated random phylogenies consisting of gapless metabolic networks under a simple model of metabolic network evolution. Second, we derived
Table 1: Values of heuristic $d_δ$ for different parent and children assignments.

<table>
<thead>
<tr>
<th>Parent</th>
<th>Left child</th>
<th>Right child</th>
<th>$d_δ$</th>
<th>Parent</th>
<th>Left child</th>
<th>Right child</th>
<th>$d_δ$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4: Operation of Algorithm 1 to solve the Gapless Minimum Mutation problem. Left: A part of a phylogenetic tree. Right: Subnetworks of nodes u, v, x, y. State at node v after a call to $\text{MinDist}$: example values of $d_f$ and reaction assignments shown at bottom right corner and on top of each reaction of v, respectively. Metabolite $m_4$ assumed to be a source. Dashed edges indicate parts of networks not shown. Distances for $r_3$: $d_δ(r_3,v) = d_δ(r_3,u) + d_δ(r_3,x) + d_δ(r_3,y) = 1 - 1 + 1 = 1$ and thus $d_f(r_3,v) = \max(\min(d_f(r_1,v),d_f(r_2,v)),0) + d_δ(r_3,v) = \max(\min(2,4),0) + 1 = 2 + 1 = 3$.

metabolic networks for 16 fungal species from the KEGG database (Kanehisa et al., 2008). We then solved the Gapless Minimum Mutation problem for both datasets.

### 3.1 Random Phylogenies

To generate a phylogenetic tree, we first started with a gapless network containing 300 random reactions from KEGG and then simulated evolution by randomly adding or removing one reaction at time. Only additions and deletions, or reaction flips, that preserved gaplessness were allowed. Probability of both the addition and deletion was 0.5. The reaction to be added or deleted was chosen uniformly from the set of reactions whose addition or deletion preserved gaplessness. After each flip, a speciation event occurred at a fixed probability 0.005 resulting in a new branch in the phylogenetic tree. The process was terminated after a tree of 30 nodes was generated resulting in networks of $249 \pm 22$ reactions at each node.

The generated taxa were reconstructed by the Fitch algorithm and Algorithm 1. Prior to reconstruction, the input taxa were randomly perturbed to simulate effects of annotation errors. Specifically, each reaction present in the taxa was deleted with the probability $p_d = 0.01, 0.02, 0.05, 0.1$. Table 2

---

We use the $\pm$ notation to indicate standard deviations.
Table 2: Reconstructing random phylogenies when errors were introduced to data by deleting reactions from taxa with a fixed probability $p_d$. Columns FitchError and GaplessError give the reconstruction error measured as the average Hamming distance between reconstructed network and generating taxa at an internal node. Columns Gaps and Fills show the average number of gapped reactions and gapfill reactions added by our algorithm. Results are averages over 25 repeats. Standard deviations given in parentheses.

<table>
<thead>
<tr>
<th>$p_d$</th>
<th>FitchError</th>
<th>GaplessError</th>
<th>Gaps</th>
<th>Fills</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>52.3 (4.12)</td>
<td>50.3 (3.80)</td>
<td>0.18 (0.11)</td>
<td>0.15 (0.09)</td>
</tr>
<tr>
<td>0.005</td>
<td>53.3 (3.49)</td>
<td>51.2 (3.47)</td>
<td>0.35 (0.22)</td>
<td>0.24 (0.14)</td>
</tr>
<tr>
<td>0.01</td>
<td>54.8 (3.93)</td>
<td>52.4 (4.09)</td>
<td>0.55 (0.21)</td>
<td>0.35 (0.13)</td>
</tr>
<tr>
<td>0.02</td>
<td>57.3 (4.84)</td>
<td>54.9 (4.60)</td>
<td>0.94 (0.35)</td>
<td>0.59 (0.20)</td>
</tr>
<tr>
<td>0.05</td>
<td>63.3 (4.06)</td>
<td>60.3 (4.04)</td>
<td>1.82 (0.60)</td>
<td>1.11 (0.27)</td>
</tr>
<tr>
<td>0.1</td>
<td>76.5 (4.14)</td>
<td>73.2 (4.00)</td>
<td>3.65 (1.00)</td>
<td>2.37 (0.53)</td>
</tr>
</tbody>
</table>

Table 3: Species in fungal dataset. Columns Reactions, Gaps and Fills give the number of all reactions and gapped reactions in the initial networks, and reactions added by the algorithm to fill gaps, respectively.

<table>
<thead>
<tr>
<th>Abbr</th>
<th>Species</th>
<th>Reactions</th>
<th>Gaps</th>
<th>Fills</th>
</tr>
</thead>
<tbody>
<tr>
<td>ago</td>
<td>Ashbya gossypii</td>
<td>272</td>
<td>64 (23.5%)</td>
<td>37 (+13.6%)</td>
</tr>
<tr>
<td>afm</td>
<td>Aspergillus fumigatus</td>
<td>468</td>
<td>80 (17.1%)</td>
<td>50 (+10.7%)</td>
</tr>
<tr>
<td>ani</td>
<td>Aspergillus nidulans</td>
<td>382</td>
<td>74 (19.4%)</td>
<td>42 (+11.0%)</td>
</tr>
<tr>
<td>aor</td>
<td>Aspergillus oryzae</td>
<td>396</td>
<td>70 (17.7%)</td>
<td>42 (+10.6%)</td>
</tr>
<tr>
<td>cgr</td>
<td>Candida glabrata</td>
<td>270</td>
<td>70 (25.9%)</td>
<td>35 (+13.0%)</td>
</tr>
<tr>
<td>cne</td>
<td>Cryptococcus neoformans</td>
<td>336</td>
<td>68 (20.2%)</td>
<td>35 (+10.4%)</td>
</tr>
<tr>
<td>dha</td>
<td>Debaryomyces hansenii</td>
<td>326</td>
<td>66 (20.2%)</td>
<td>34 (+10.4%)</td>
</tr>
<tr>
<td>fgr</td>
<td>Fusarium graminearum</td>
<td>474</td>
<td>86 (18.1%)</td>
<td>43 (+9.1%)</td>
</tr>
<tr>
<td>kla</td>
<td>Kluyveromyces lactis</td>
<td>292</td>
<td>68 (23.3%)</td>
<td>36 (+12.3%)</td>
</tr>
<tr>
<td>mgr</td>
<td>Magnaporthe grisea</td>
<td>390</td>
<td>74 (19.0%)</td>
<td>41 (+10.5%)</td>
</tr>
<tr>
<td>ncr</td>
<td>Neurospora crassa</td>
<td>416</td>
<td>74 (17.8%)</td>
<td>38 (+9.1%)</td>
</tr>
<tr>
<td>dpch</td>
<td>Phanerochaete chrysosporium</td>
<td>410</td>
<td>90 (22.0%)</td>
<td>44 (+10.7%)</td>
</tr>
<tr>
<td>sce</td>
<td>Saccharomyces cerevisiae</td>
<td>332</td>
<td>80 (24.1%)</td>
<td>36 (+10.8%)</td>
</tr>
<tr>
<td>spo</td>
<td>Schizosaccharomyces pombe</td>
<td>278</td>
<td>68 (24.5%)</td>
<td>35 (+12.6%)</td>
</tr>
<tr>
<td>uma</td>
<td>Ustilago maydis</td>
<td>296</td>
<td>76 (25.7%)</td>
<td>47 (+15.9%)</td>
</tr>
<tr>
<td>yli</td>
<td>Yarrowia lipolytica</td>
<td>362</td>
<td>92 (25.4%)</td>
<td>43 (+11.9%)</td>
</tr>
</tbody>
</table>

3.2 Fungal Phylogenies

To experiment with a more real-world scenario, we constructed metabolic networks for 16 fungal species corresponding to 17 carbohydrate metabolism pathways (Kanehisa et al., 2008) from KEGG gene-reaction links. As shown in Table 3, the process resulted in a high number of gapped reactions in these initial networks largely due to incomplete annotations and stoichiometry in KEGG enzymes and reactions.

As input, we provided our algorithm with the initial networks and a phylogenetic tree of the species ((Fitzpatrick et al., 2006), Figure 5). Because we had gapped networks to begin with, we added a new internal node for each species and an edge to the node as described earlier.

Many microorganisms and free living fungi in particular can synthesize all their cellular components from inorganic sources, given a source of energy and
carbon such as glucose (Deacon, 2006). However, many fungi require one or more vitamins such as thiamine or biotin. To model the fungal metabolic environment, a fixed set of 1518 source metabolites were used containing fungal energy and carbon sources, cofactors such as ATP and NAD and metabolites needed to account for the large number of otherwise isolated subgraphs in the KEGG universal metabolic network.

Table 3 lists the number of reactions added to fill gaps in the internal nodes corresponding to species. On the average, 39.4 ± 3.9 reactions were used to fill 72.5 ± 7.0 gaps divided into 7.2 ± 0.9 graph components at each node. The optimal minimum mutation cost was 1082; our algorithm achieved gapless minimum mutation cost of 1789.

To give an example of how the GMM result can provide insight into metabolic network evolution and aid reconstruction curation efforts, Figure 6 shows five reactions from the internal node N22 that is the parent of species nodes mgr and ncr (Figure 5). Two reactions (KEGG ids R02933 and R02957) remained gapped after the first pass. Three reactions were predicted by the algorithm to fill these particular gaps (R01184, R01481, R01483). All filling reactions were used in parent but were absent from children, thus addition of each reaction increased the parsimony cost by one.

Even though KEGG reaction-gene links were missing for reactions R01184 and R01481, algorithm predictions were supported by homologous genes (Arvas et al., 2007) found for both reactions in M. grisea. Further, a homologous gene was found also in N. crassa for reaction R01184. No gene was found to support the predicted existence of reaction R01483, warranting further study. For the two gapped reactions, homologues were found in both organisms, supporting KEGG data.

4 CONCLUSIONS

In this paper, we introduced a maximum parsimony algorithm for reconstructing gapless ancestral metabolic networks for a given phylogenetic tree. Furthermore, the method can be used to suggest gapless variants of draft metabolic networks of observed species given as input. The algorithm minimizes the number of mutations in the phylogenetic tree while...
ACKNOWLEDGEMENTS

We would like to thank Pasi Rastas and Esko Ukkonen for insightful discussions. This work was financially supported by Academy of Finland grant 118653 (ALGODAN), in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886-PASCAL2, and by the Academy of Finland postdoctoral researcher’s fellowship 127715 (as part of the Finnish Centre of Excellence in White Biotechnology - Green Chemistry, Project No. 118573). This publication only reflects the authors’ views.

REFERENCES


