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

Pitkänen E., Rousu J. and Arvas M..
MINIMUM MUTATION ALGORITHM FOR GAPLESS METABOLIC NETWORK EVOLUTION.
DOI: 10.5220/0003132200280038
In *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms* (BIOINFORMATICS-2011), pages 28-38
ISBN: 978-989-8425-36-2

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 $\mathcal{R}$, is in the network. We use the shorthand $r_i \in N$ when $N_i = 1$. Further we assume a set of metabolites $\mathcal{M}$ is consumed and produced by the m reactions. The set of substrate and product metabolites of a reaction r_i are given by $S(r_i) \subset \mathcal{M}$ and $P(r_i) \subset \mathcal{M}$, 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 $S \subseteq \mathcal{M}$ 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 $b \in \mathcal{M}$ 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

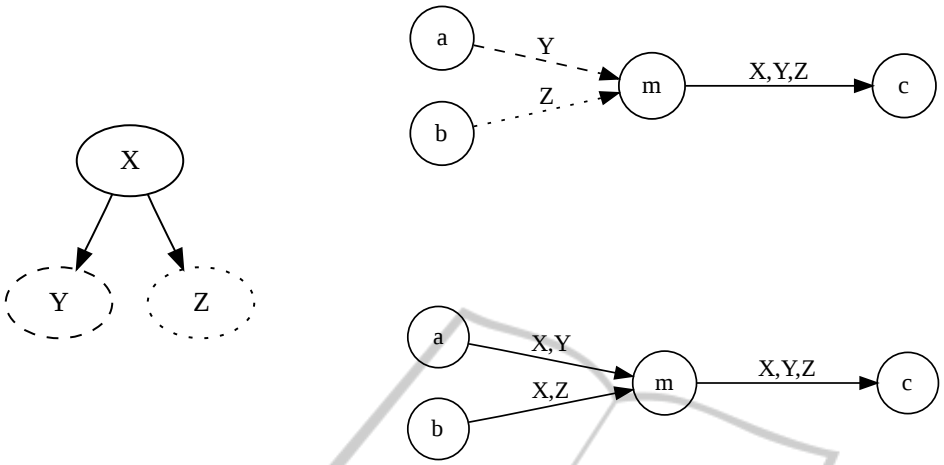

Figure 1: Left: a small example phylogeny for three metabolic networks X, Y, Z. Right: two parsimonious scenarios involving metabolic pathways $a \rightarrow m, b \rightarrow m$ and $m \rightarrow c$. Metabolites and pathways are shown as circles and arrows, respectively. Pathways are labeled with organisms which have the pathway. In upper scenario, only pathway $m \rightarrow c$ is assigned to ancestor X, thus leaving metabolite m without a producing pathway. In lower scenario, both pathways from Y and Z are assigned to X.

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 (Pitkänen 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. In the Minimum Set Cover problem, we ask for the smallest subset $C' \subseteq C$ 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 $\mathcal{R}$ to 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

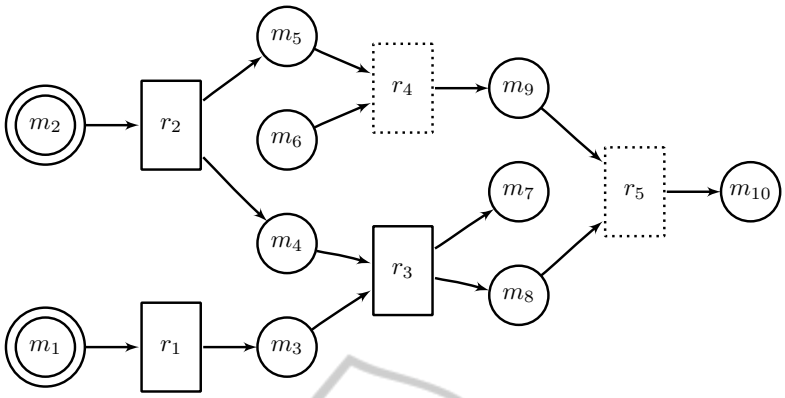

Figure 2: Example metabolic network with reactions $N = \{r_1, \ldots, r_5\}$ and metabolites $\{m_1, \ldots, m_{10}\}$. When $S = \{m_1, m_2\}$ (double circles), reactions r_4 and r_5 (dotted rectangles) are reaction gaps in N. However, the network N under $S' = \{m_1, m_2, m_6\}$ would be gapless, because then reaction r_4 would be reachable. On the other hand, $N' = \{r_1, r_2, r_3\}$ is gapless under $S = \{m_1, m_2\}$.

be $S = \{a_j \mid c_j \in C\}$.

A minimal solution to GMM will contain at node v_3 a network with r_i and x, and a minimal number of reactions q_j 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. Figure 3 shows the reduction from an example set cover instance $X = \{x_1, \ldots, x_5\}$ and $c_1 = \{x_1, x_2, x_3\}, c_2 = \{x_1, x_2, x_4\}, c_3 = \{x_3, x_5\}$.

Since the problem is both in NP and NP-hard, the claim follows. □

Theorem 2.4. *Gapless Minimum Mutation cannot be efficiently* α*-approximated unless* $P = NP$.

Proof. We show next that the reduction described above is actually an approximation-ratio preserving reduction and we can thus exploit the inapproximability of the set cover problem. Set cover problem has been shown to be hard to approximate within a logarithmic factor unless $P = NP$ (Raz and Safra, 1997; Alon et al., 2006).

Assume that there is an α-approximation algorithm for GMM for some $\alpha > 1$. We can thus obtain a solution of cost $\leq \alpha OPT$ where OPT is an optimal solution cost of GMM. Given a set cover instance with optimal size k, we obtain a GMM instance of optimal cost $2k$ in polynomial time with the above reduction. Solving the instance approximatively we get a solution of cost at most $2\alpha k$. This yields an approximative solution $\hat{k} \leq \alpha k$ 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 $L(u') \leftarrow L(u)$. 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.

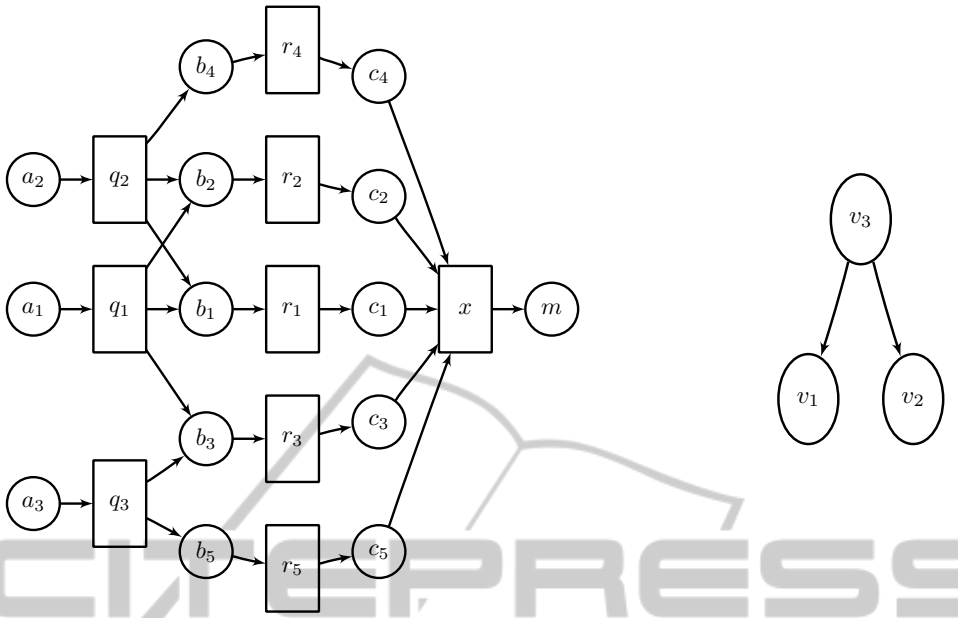

Figure 3: Example reduction of a Minimum Set Cover instance with $X = \{x_1, \ldots, x_5\}$ and $c_1 = \{x_1, x_2, x_3\}, c_2 = \{x_1, x_2, x_4\}$ and $c_3 = \{x_3, x_5\}$ to Gapless Minimum Mutation problem. Left: reactions in $\mathcal{R}$. Right: tree T.

Algorithm 1: `GaplessMinimumMutation`: Local adjustment algorithm for gapless minimum mutation.

1: Input: tree T, taxa L
2: $F \leftarrow \texttt{Fitch}(T, L)$
3: **for all** internal nodes v in top-down order **do**
4: **if** v is root **then**
5: **for all** $i \in \{1, \ldots, m\}$ **do**
6: $L_i(v) \leftarrow 0$ if $0 \in F_i(v)$; otherwise $L_i(v) \leftarrow 1$
7: **else**
8: **for all** $i \in \{1, \ldots, m\}$ **do**
9: **if** $F_i(v) = \{0, 1\}$ **then**
10: $L_i(v) \leftarrow L_i(pa(v))$
11: **else**
12: $L_i(v) \leftarrow 0$ if $0 \in F_i(v)$; otherwise $L_i(v) \leftarrow 1$
13: $(D, M) \leftarrow \texttt{MinDist}(v, S)$
14: $L(v) \leftarrow L(v) \cup \texttt{MinFill}(v, S, D, M)$

In Algorithm 1 at line 2, we first compute *equality sets* $F_i(v) \subseteq \{0, 1\}$ 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

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

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 Γ. 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 Γ 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 reachable to v:

$$d_f(v, r_i) = \max_{m \in S(r_i)} \min_{q \in Pr(m)} d_f(v, q) + d_\delta(v, r_i)$$

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

$G_i(w,v) = F_i(w)$ iff $w \in ch(v)$, where the parent and children of v are denoted by $pa(v)$ and $ch(v)$, respectively. Cost function $d_\delta(v,r_i)$ specifies the increase in parsimony cost when adding reaction r_i only and is defined as $d_\delta(v,r_i) = \delta(v,pa(v),r_i) + \sum_{c\in ch(v)} \delta(v,c,r_i)$ with

$$\delta(v,w,r_i) = \begin{cases} 1 & \text{if } G_i(w,v) = \{0\} \text{ and } L_i(v) = 0 \\ -1 & \text{if } G_i(w,v) = \{1\} \text{ and } L_i(v) = 0 \\ 0 & \text{if } G_i(w,v) = \{0,1\} \text{ and } L_i(v) = 0 \\ 0 & \text{if } L_i(v) = 1 \end{cases}$$

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

Parsimony cost increase d_δ at each reaction is always non-negative. To see this, we can enumerate the values for d_δ 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_f, ensuring that distances strictly increase when traversing the network.

Algorithm 2: `MinFill`.

```
1: Input: v,S,D,M
2: Γ ← ∅; Q ← {r ∈ v | D(r) = ∞}
3: while |Q| > 0 do
4:    r ← pop(Q)
5:    for all o ∈ S(r) do
6:       if o ∉ M then
7:          q ← argmin_{q∈Pr(o)} D(q)
8:          if D(q) = ∞ then
9:             return "Impossible to find gapfilling
               set"
10:         if q ∉ Γ and q ∉ v then
11:            push(Q,q)
12:            Γ ← Γ ∪ {r}
13: Return Γ
```

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

Algorithm 3: `MinDist`: compute distances $d_f(v,S)$.

```
1: Input: v,S
2: M ← S; Q ← min-heap
3: for all r ∈ {r ∈ R | S(r) ⊆ S} do
4:    insert(Q,r,0)
5: for all r ∈ R do
6:    D(r) ← 0 if r ∈ Q; D(r) ← ∞ otherwise
7: while |Q| > 0 do
8:    r ← extract-min(Q)
9:    for all o ∈ P(r) do
10:      if o ∉ M then
11:         M ← M ∪ {o}
12:         for all q ∈ {q ∈ R | D(q) = ∞ ∧ S(q) ⊆ M}
            do
13:            D(q) ← D(r) + δ_f(v,q) + ε
14:            insert(Q, q, D(q))
15: return (D,M)
```

`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.

Parent	Left child	Right child	d_δ	Parent	Left child	Right child	d_δ
0	0	0	3	1	0	0	1
0	0	1	1	1	0	1	**0**
0	0	0,1	2	1	0	0,1	0
0	1	1	**0**	1	1	1	**0**
0	1	0,1	**0**	1	1	0,1	**0**
0	0,1	0,1	1	1	0,1	0,1	**0**

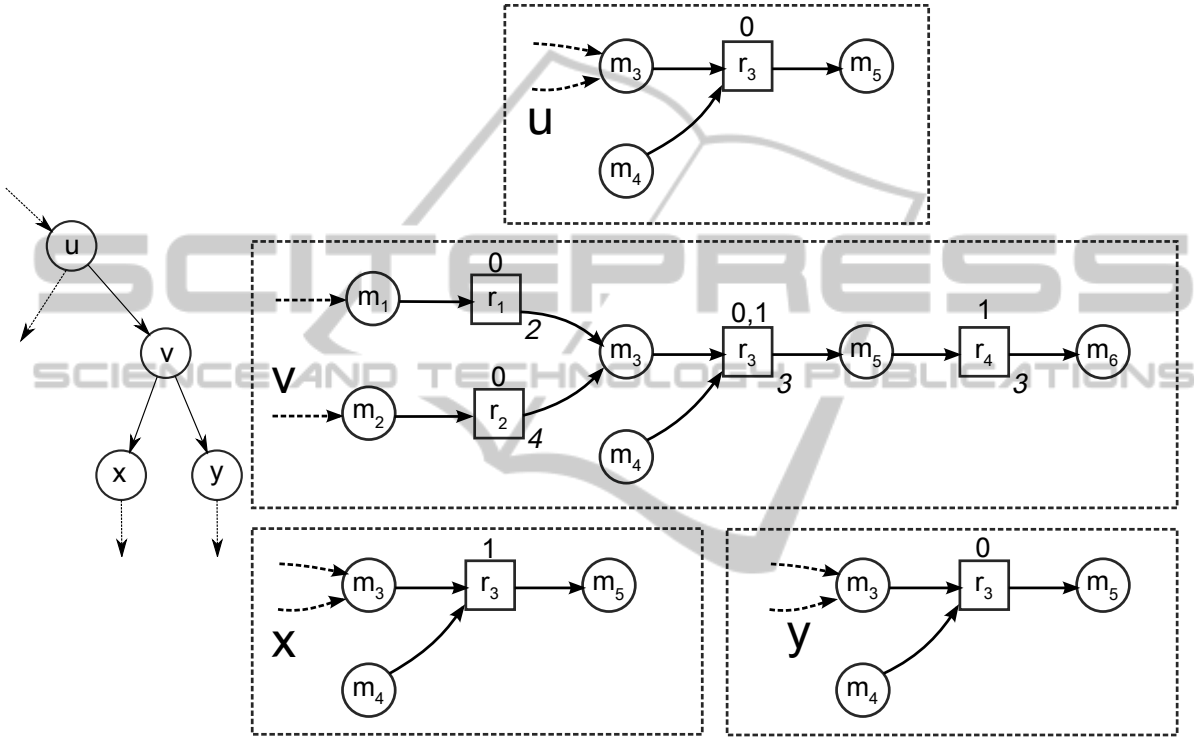

Figure 4: Operation of Algorithm 1 to solve the Gapless Minimum Mutation problem. Left: A part of a phylogenetic tree. Right: Subnetworks of nodes u,v,x,y. State at node v after a call to `MinDist`: example values of d_f and reaction assignments shown at bottom right corner and on top of each reaction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

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 occured 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 ± 22 reactions at each node[1].

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 $p_d = 0, 0.01, 0.02, 0.05, 0.1$. Table 2

[1]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.

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

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.

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

shows the reconstruction error measured as the average Hamming distance between node labels in reconstruction and generating taxa. Further, the average number of gapped and gapfill reactions at each node are shown. For instance, with random deletion probability $p_d = 0.1$, the reconstruction errors measured, 76.5 and 73.2 for Fitch's and our algorithm, respectively, were statistically different from each other (paired $t(48) = 2.87$, $p = 0.006$). Moreover, the simulated trees contained on the average 3.65 gaps in each network. Our algorithm added 2.37 gap-filling reactions to each network on the average. Regardless of parameter p_d, gapless reconstruction of each instance took about 27 seconds on a standard desktop computer running a Python implementation our algorithm.

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

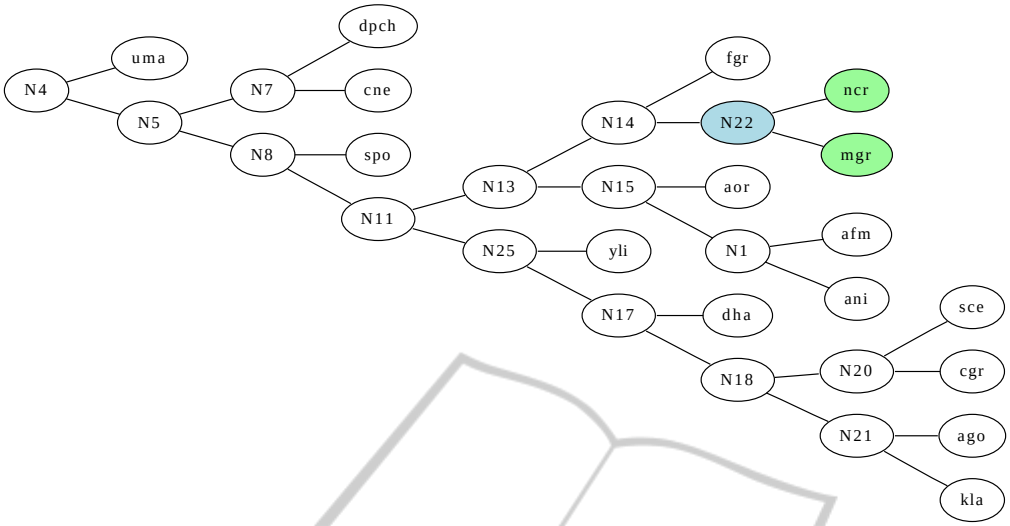

Figure 5: A phylogeny tree for the 16 fungal species rooted at node N4. Shaded nodes indicate the two species and their immediate ancestor N22 highlighted in Figure 6.

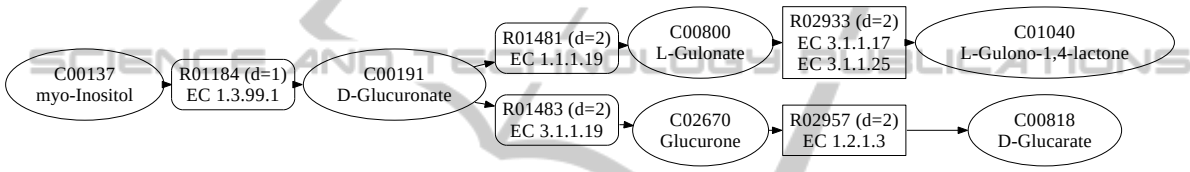

Figure 6: A subnetwork of the metabolic network reconstructed at node N22 of the phylogenetic tree shown in Figure 5. Reactions and metabolites are drawn as rectangles and ellipses, respectively. Rounded rectangles show the three reactions added by the algorithm to fill the gapped reactions R02933 and R02957. Distances d_f given in parentheses as d.

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

maintaining the gaplessness of the ancestral networks. Thus, the algorithm can be used both to elucidate network structures in ancestral nodes and to fill in gaps in draft metabolic networks in a evolutionarily plausible manner.

We argue that such approach, where the reconstruction networks are required to be gapless, improves prediction performance over the unconstrained case. This work extends the method of (Pitkänen et al., 2008), where gapless reconstructions of individual metabolic networks were inferred from sequence data, to take into account the phylogenetic context of the reconstructed network.

The proposed algorithm was found to perform well in practice despite the computational complexity of the underlying problem. In experiments with random data, the algorithm was able to recover the original data from perturbed input more accurately than the baseline method that did not enforce gaplessness in reconstructed networks. Moreover, the algorithm yielded explanations to the question why a given reaction (enzyme) appears in an ancestral network by suggesting the required reactions that render the reaction gapless. This is especially important when we attempt to uncover the evolutionary history leading into observed networks.

While we experimented only with a simple random model of evolution, the framework introduced here lends itself to more realistic models.

Exploring this direction is left as future work, though we note the importance of incorporating sequence data with the joint metabolic network/phylogenetic tree topology driven approach presented here. To this end, dealing with inaccuracies and omissions in the underlying metabolic reaction databases presents an additional challenge.

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

Alon, N., Moshkovitz, D., and Safra, S. (2006). Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2(2):153–177.

Arvas, M., Kivioja, T., Mitchell, A., Saloheimo, M., Ussery, D., Penttilä, M., and Oliver, S. (2007). Comparison of protein coding gene contents of the fungal phyla *Pezizomycotina* and *Saccharomycotina*. *BMC Genomics*, 8(1):325.

Borenstein, E., Kupiec, M., Feldman, M. W., and Ruppin, E. (2008). Large-scale reconstruction and phylogenetic analysis of metabolic environments. *PNAS*, 105(38):14482–14487.

Bourque, G. and Sankoff, D. (2004). Improving gene network inference by comparing expression time-series across species, developmental stages or tissues. *J Bioinform Comput Biol*, 2(4):765–783.

Caetano-Anollés, G., Yafremava, L., Gee, H., Caetano-Anollés, D., Kim, H., and Mittenthal, J. (2009). The origin and evolution of modern metabolism. *The International Journal of Biochemistry & Cell Biology*, 41(2):285–297.

Clemente, J., Satou, K., and Valiente, G. (2007). Phylogenetic reconstruction from non-genomic data. *Bioinformatics*, 23(2):e110.

Clemente, J. C., Ikeo, K., Valiente, G., and Gojobori, T. (2009). Optimized ancestral state reconstruction using sankoff parsimony. *BMC Bioinformatics*, 10(51).

Dandekar, T., Schuster, S., Snel, B., Huynen, M., and Bork, P. (1999). Pathway alignment: application to the comparative analysis of glycolytic enzymes. *Biochem J.*, 343(Pt 1):115–124.

Deacon, J. (2006). *Fungal biology*. Wiley-Blackwell.

Fitch, W. M. (1971). Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Zool.*, 20:406–416.

Fitzpatrick, D., Logue, M., Stajich, J., and Butler, G. (2006). A fungal phylogeny based on 42 complete genomes derived from supertree and combined gene analysis. *BMC Evolutionary Biology*, 6(1):99.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press.

Handorf, T., Christian, N., Ebenhöh, O., and Kahn, D. (2008). An environmental perspective on metabolism. *Journal of Theoretical Biology*, 252(3):530–537.

Jamshidi, N. and Palsson, B. O. (2007). Investigating the metabolic capabilities of *Mycobacterium tuberculosis* H37Rv using the in silico strain iNJ661 and proposing alternative drug targets. *BMC Systems Biology*, 1(26).

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., and Yamanishi, Y. (2008). Kegg for linking genomes to life and the environment. *Nucleic Acids Res.*, 36:D480–D484.

Kreimer, A., Borenstein, E., Gophna, U., and Ruppin, E. (2008). The evolution of modularity in bacterial metabolic networks. *Proceedings of the National Academy of Sciences*, 105(19):6976.

Lacroix, V., Cottret, L., Thebault, P., and Sagot, M.-F. (2008). An introduction to metabolic networks and their structural analysis. *IEEE Transactions on Computational Biology and Bioinformatics*, 5(4):594–617.

Mano, A., Tuller, T., Bj, O., and Pinter, R. Y. (2010). Comparative classification of species and the study of pathway evolution based on the alignment of metabolic pathways. *BMC Bioinformatics*, 11(Suppl 1):S38.

Mithani, A., Preston, G., and Hein, J. (2010). A bayesian approach to the evolution of metabolic networks on a phylogeny. *PLoS Computational Biology*, 6(8).

Mithani, A., Preston, G. M., and Hein, J. (2009). A stochastic model for the evolution of metabolic networks with neighbor dependence. *Bioinformatics*, 25(12):1528–1535.

Palsson, B. (2006). *Systems biology: properties of reconstructed networks*. Cambridge University Press Cambridge.

Pitkänen, E., Rantanen, A., Rousu, J., and Ukkonen, E. (2005). Finding feasible pathways in metabolic networks. In *Advances in Informatics: 10th Panhellenic Conference on Informatics (PCI 2005). Lecture Notes in Computer Science 3746*, pages 123–133.

Pitkänen, E., Rantanen, A., Rousu, J., and Ukkonen, E. (2008). A computational method for reconstructing gapless metabolic networks. In *Proceedings of the 2nd International Conference on Bioinformatics Research and Development (BIRD'08)*, volume 13 of *Communications in Computer and Information Science*. Springer.

Pitkänen, E., Rousu, J., and Ukkonen, E. (2010). Computational methods for metabolic reconstruction. *Current Opinion in Biotechnology*, 21(1):70–77.

Raman, K. and Chandra, N. (2009). Flux balance analysis of biological systems: applications and challenges. *Briefings in Bioinformatics*, 10(4):435–449.

Rantanen, A., Rousu, J., Jouhten, P., Zamboni, N., Maaheimo, H., and Ukkonen, E. (2008). An analytic and systematic framework for estimating metabolic flux ratios from 13 C tracer experiments. *BMC Bioinformatics*, 9(1):266.

Raz, R. and Safra, S. (1997). A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 475–484.

Sankoff, D. (1975). Minimal mutation trees of sequences. *SIAM J. Appl*, 28:35–42.

Sharan, R. and Ideker, T. (2006). Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24:427–433.

Sigurdsson, M. I., Jamshidi, N., Jonsson, J. J., and Palsson, B. O. (2009). Genome-scale network analysis of imprinted human metabolic genes. *Epigenetics*, 4(1):43–46.

Tohsato, Y., Matsuda, H., and Hashimoto, A. (2000). A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 376–383.

Tuller, T., Birin, H., Gophna, U., Kupiec, M., and Ruppin, E. (2010). Reconstructing ancestral gene content by coevolution. *Genome Res.*, 20(1):122–132.

Wagner, A. (2009). Evolutionary constraints permeate large metabolic networks. *BMC Evolutionary Biology*, 9(1):231.