

A computational method for reconstructing gapless metabolic networks

Esa Pitkänen^{1,3}, Ari Rantanen², Juho Rousu¹, and Esko Ukkonen¹

¹ Department of Computer Science, University of Helsinki, Finland

² Institute of Molecular Systems Biology, ETH Zürich, Switzerland

³ Corresponding author, e-mail: esa.pitkanen@cs.helsinki.fi

Abstract. We propose a computational method for reconstructing metabolic networks. The method utilizes optimization techniques and graph traversal algorithms to discover a set of biochemical reactions that is most likely catalyzed by the enzymatic genes of the target organism. Unlike most existing computational methods for metabolic reconstruction, our method generates networks that are structurally consistent, or in other terms, gapless. As many analyses of metabolic networks, like flux balance analysis, require gapless networks as inputs, our network offers a more realistic basis for metabolic modelling than the existing automated reconstruction methods. It is easy to incorporate existing information, like knowledge about experimentally discovered metabolic reactions or metabolites into the process. Thus, our method can be used to assist in the manual curation of metabolic network models as it is able to suggest good candidate reactions for filling gaps in the existing network models. However, it is not necessary to assume any prior knowledge on composition of complete biochemical pathways in the network. We argue that this makes the method well-suited to analysis of organisms that might differ considerably from previously known organisms. We demonstrate the viability of our method by analysing the metabolic network of the well-known organism *Escherichia coli*.

1 Introduction

Structural models of cellular metabolism have proven to be very successful in answering many relevant biological research questions. The global organization of the metabolism can be discovered from the structural models [10]. In *constraint based modelling* of the metabolism [5, 32, 35–37, 27], phenotypic behaviour, robustness and metabolic capabilities of an organism are analysed based on the structural models.

The process of creating a metabolic network model corresponding to the genome of the organism under study is called *metabolic reconstruction* [17]. In order to reconstruct a structural model of metabolism a complete set of metabolic reactions operating in the organism has to be discovered. These reactions define the topology of an intertwined *metabolic network*, as the reactions are connected to each other by common substrate and product metabolites.

The most common method for reconstructing the structural model of a metabolic network is based on the combination of functional annotation of metabolic genes in the target organism by sequence similarity and manual utilization of literature information [13]. In the first technique, the functions of known enzymatic genes in a database are assigned to homologous genes in the target organism, thus adding the corresponding enzymatic reactions to the reconstructed metabolic network [17]. Knowledge on metabolic function annotation can also be derived from other sources, such as chromosome clustering [18], detection of protein fusion events [29], occurrence profiles [39], phylogenic profiles [7], and regulons [30]. The metabolic network is then curated by manually adding experimentally verified reactions and by fixing the observed inconsistencies.

A *reaction gap* is the most common example of an inconsistency in the reconstructed metabolic network [31]. A reaction gap occurs, when substrates of an internal reaction cannot be produced from the external substrates of the network. In constraint based modelling of metabolism, reaction velocities, or fluxes, in the metabolic network are explicitly modelled. As pathways with gaps cannot carry any flux in steady state conditions, the most accurate results are obtained when the analysis techniques are applied with gapless models.

Manual curation and gap-filling of genome-scale metabolic networks consisting of hundreds, even thousands of reactions is, however, very time-consuming and error-prone [16]. But, because of the indisputable usefulness metabolic network models in system-wide biological studies, manual curation of metabolic networks is considered to be worthwhile for the ever increasing number of organisms ranging from simple bacteria to mammalian cells [6, 24, 12, 13, 11, 34]. Unfortunately, most reconstructed models still contain some gaps even after the manual curation.

Many computational methods and tools have been developed to assist the metabolic reconstruction and curation of genome-scale metabolic network models. For instance, the Pathway Tools software reconstructs metabolic networks by first determining how large portion of the enzymes of each pathway in a pathway database are present in the organism [23]. This is done by comparing a list of EC numbers given as input to the program against the EC numbers in the MetaCyc pathway database. EC numbers specify a functional classification of enzymatic activity [22]. Then the presence of each pathway is predicted. Roughly, the more enzymes appear to be present, the more confident we are that the pathway exists.

As a drawback, Pathway Tools cannot recreate truly novel subnetworks, as it is limited to the pathways in its database. The studies of central pathways such as TCA cycle, pentose phosphate pathway and glycolysis in microbial metabolic networks have indicated that the structure of these pathways varies in many organisms, from the textbook definition [9]. Thus, it is useful not to let hard-coded definitions of pathway structures based on previous knowledge affect the prediction of pathways in excess in newly sequenced organisms. In addition, Pathway Tools incorporates a second phase, where gaps left from the first phase are filled. The method is called the Pathway Hole Filler, which is a Naive Bayesian clas-

sifier combining evidence from genome structure such as chromosome clustering and others [19]. It predicts for each enzyme that was missing from the original reconstruction whether it should be added to the metabolic network.

Recently, an optimization based method called GapFill for filling gaps in a draft metabolic network was introduced by Kumar et al. [28]. In the method gaps are first discovered, then filled one by one by adding reactions to the network or by adding reverse directions for unidirectional reactions in the model (see Section 4 for a more detailed comparison to the present method).

It has been observed that reconstruction benefits from having multiple data sources combined [25]. In the approach of [26], experimental data is combined with the knowledge on metabolic network structure. By assuming that the genes encoding for enzymes on the same pathway are co-regulating, they try to find similarly expressed genes which could then be annotated with enzymatic functions on the same pathway. In [21], a draft metabolic network is automatically curated by minimizing the difference between the experimentally determined metabolic fluxes and the fluxes estimated by the flux balance analysis.

In the present paper, we introduce a new computational method for assisting in the task of metabolic reconstruction. In the method, optimization techniques and graph traversal algorithms are utilized to find a set of biochemical reactions that is most likely catalyzed by the genes of the target organism. The resulting network is guaranteed to be gapless, that is, each reaction in the reconstructed network is connected with a feasible metabolic pathway to external source metabolites, such as glucose. Thus, our method is able to provide feasible suggestions that are backed by the genomic evidence about the topology of the reconstructed metabolic network. This should significantly speed up the curation of metabolic network models. Our method does not assume any existing knowledge on specific pathways. In other words, pathway collections or logical rules coding pathway information are not needed. However, it is possible to utilise easily information about known reactions and metabolites to aid in the reconstruction process. For example, knowledge about experimentally observed metabolites and reactions can be easily exploited in the framework. Thus, the framework can be used both to produce *ab initio* reconstructions from the genome sequence data, as well as to fill the gaps in a draft model of a metabolic network.

To the authors' best knowledge, the presented method is the only computational technique for metabolic reconstruction that integrates the requirement for structural consistency and the search of reactions that are most likely to be catalyzed by the enzymes of the target organism to a single, global optimization task.

2 Methods

We formulate the metabolic reconstruction problem formally in the optimization context, a setting popular in the analysis of reconstructed metabolic networks [5, 32]. We start by introducing first the key concept of reaction reachability, and

then define metabolic reconstruction as the problem of finding a set of reachable reactions that is likely catalyzed by the genes of the target organism. Finally, a formulation in a mixed integer linear programming framework to solve the problem is given.

2.1 Reaction reachability

Informally, a reaction gap occurs, when the metabolic network model is unable to supply all substrates of some reaction in the network. The model obviously contains an error: either the reaction in question should be removed, or the network should be modified to be able to provide the substrates.

We give a formulation of this idea by looking at reachability in an *and-or-graph* corresponding to the metabolic network under study [33]. In this context, we relate reactions with and-nodes and metabolites with or-nodes. Particularly, a reaction $r_i = (I_i, P_i)$ is specified by its substrates I_i and products P_i . We then investigate whether reactions can be reached in the network using the following rules, given network input metabolites A . The input metabolites A correspond to the nutrients of the organism under study. A typical example of inputs would include glucose as the carbon source.

- A reaction $r_i = (I_i, P_i)$ is *reachable* from A in R , if each metabolite in I_i is reachable from A in R .
- A metabolite m is *reachable* from A in R , if $m \in A$ or some reaction $r_j = (I_j, P_j)$ such that $m \in P_j$ is reachable from A in R .

We want all reactions in the reconstructed network to be reachable. This corresponds to not having any reaction gaps in the network.

Definition 1 (Feasible metabolic network). *A metabolic network consisting of reactions R is called feasible with respect to the source metabolites A , if and only if all reactions $r \in R$ are reachable from A in R .*

We will now formulate the discovery of the largest feasible network consisting of reactions in R with respect to source metabolites A as a linear programming problem. This formulation will be then utilized in the metabolic reconstruction. By v_i we denote the rate, or *flux* of reaction r_i in the network. In other words, v_i models the activity of reaction r_i . Now, the reachability of metabolites in the network can be coded by two constraints on their fluxes. The first constraint requires that the production of each metabolite not in A is equal or greater than its consumption,

$$\sum_i \delta_{ij} v_i - t_j \geq 0, \tag{1}$$

where δ_{ij} is -1 , 1 or 0 depending on whether reaction r_i consumes, produces or does not use metabolite m_j , $v_i \geq 0$ is the rate of reaction r_i and $t_j \geq 0$ is the rate of the dilution reaction of metabolite m_j . The dilution reactions are required to ensure that there will be no cycles which are not connected with a path to network inputs A .

This is achieved by constraints requiring that whenever a metabolite m_j is produced by at least one reaction, the flux of the dilution reaction for that metabolite must be greater than zero,

$$t_j \geq \alpha \sum_{i \in P(m_j)} v_i, \quad (2)$$

where $P(m_j)$ is the set of reactions producing metabolite m_j and $0 < \alpha < 1$.

To see that metabolite dilution reactions ensure that there will be no cycles which are not connected to network inputs A , consider a cycle with reaction rates $v_i > 0$. Then, as the influx of some metabolite m_j would be greater than zero, also the corresponding dilution reaction flux $t_j > 0$. Since this dilution reaction participates to the steady state constraint of metabolite m_j , only a part of the flux to m_j can be used in fluxes leaving m_j . Thus, as all metabolites are constrained by (1), all reaction rates v_i (and t_j) in the cycle have to be zero.

We can find all reachable reactions in the network by maximising $\sum_{r_i \in R} v_i$ under the above constraints: the reaction r_i is reachable if and only if $v_i > 0$.

2.2 Metabolic reconstruction problem

Next we formulate metabolic reconstruction as a mixed integer linear programming problem involving the choice of reachable reactions to maximise a given score function over the reactions.

Problem 1 (Metabolic reconstruction). Given a set of reactions \mathcal{R} , a set of input metabolites A , a threshold value b , and a score function $f_b : \mathcal{R} \rightarrow \mathbb{R}$, find a subset R of \mathcal{R} such that

1. $F_b(R) = \sum_{r \in R} f_b(r)$ is maximized and
2. the metabolic network with reactions R is feasible with respect to inputs A .

The corresponding mixed integer linear programming problem is the following.

Problem 2 (Metabolic reconstruction MILP).

$$\max_{\mathbf{x}} \sum_{r_i} f_b(r_i) x_i$$

such that

$$\frac{1}{N} x_i \leq v_i \quad (3)$$

$$v_i \leq M x_i, \quad (4)$$

$$\sum_i \delta_{ij} v_i - t_j \geq 0, \quad (5)$$

$$t_j \geq \alpha \sum_{r_i \in P(m_j)} v_i \text{ and} \quad (6)$$

$$x_i \in \mathbb{N} \quad (7)$$

where $x_i \in \{0, 1\}$ specifies whether reaction r_i is included in the result, N and M are appropriately large numbers, $0 < \alpha < 1$, v_i is the rate of reaction r_i , t_j is the dilution reaction rate corresponding to metabolite m_j and $P(m_j)$ is the set of reactions producing metabolite m_j .

Values for x_i , v_i and t_j are chosen during the optimization, while N , M and α are constants. Constraints (3) and (4) require that $x_i = 0 \Leftrightarrow v_i = 0$. Reaction cycles not connected to network inputs are disallowed with constraints (5) and (6).

2.3 Reaction and network scores

We derive scores f_b for reactions from the sequence homology evidence from the genome of the organism under study as follows. The score $f_b(r)$ for the reaction r is the homology score corresponding to the best match for a sequence in the target genome and a sequence which already has reaction r as its functional annotation. In other words, scores f_b can be calculated with

$$f_b(r) = \max_{s \in G} \max_{t \in U_r} S(s, t) - b, \quad (8)$$

where s is a sequence in the genome G of the organism under study and t is a sequence chosen from the set of sequences U_r annotated with reaction r in the protein database. Function $S(s, t)$ gives the degree of sequence similarity for sequences s and t ; in this study, we use BLAST [2], the mainstay sequence search tool in bioinformatics.

We score a network simply as the sum of scores of reactions R of the network,

$$F_b(R) = \sum_{r \in R} f_b(r).$$

The threshold parameter $b > 0$ specifies a value dividing reactions into two groups. Reactions with a positive score $f_b(r)$ bring positive contribution to the overall network score $F_b(R)$. Thus, if we would not require that all reactions are reached from inputs, the optimization process would add all such reactions to the result, while leaving every reaction with a negative score out. In particular, the optimization may include a reaction with $f_b(r) < 0$ to the network only if the addition makes possible with respect to the feasibility constraint to add also reactions giving positive contribution to the score.

2.4 Divide-and-conquer approach

Unfortunately, the exact formulation of the above reconstruction problem is computationally infeasible for genome-scale instances. To tackle with the complexity, we revise the formulation by dividing the original problem into smaller subproblems and solving the subproblems individually. By this heuristic divide-and-conquer approach, we are able to solve realistic genome-scale instances. The central idea is to first find a good acyclic path that will be augmented into a feasible pathway, and repeat the process until we are satisfied with the outcome.

We first generate a random acyclic path $P = (r_1, \dots, r_n)$, $r_i \in \mathcal{R}$, starting from a randomly chosen source metabolite $a \in A$. The next reaction r_{i+1} on the path is selected by taking a random reaction that consumes a random product of r_i while ensuring that the path stays acyclic. Additionally, we require that $d_R(A, r_{i+1}) > d_R(A, r_i)$ for all reactions on the path, where $d_R(A, r)$ is the *production distance* from metabolites A to reaction r in the network R [33]. The production distance is the smallest diameter taken over all minimal feasible pathways from A to r , while diameter of a pathway is the length of the longest acyclic path in the pathway. In effect, the distance $d_R(A, r)$ is the minimum number of successive reactions needed to convert metabolites A into products of the reaction r . If $d_R(A, r)$ is defined, then there exists a feasible pathway connecting A to r . Finally, we take the longest path $P' = (r_1, \dots, r_k)$, $1 \leq k \leq n$, such that the path score is positive, $\sum_{1 \leq i \leq k} f_b(r_i) > 0$.

We then augment this initial path P' by adding all acyclic paths which start from sources A and end in some reaction $r \in P'$, requiring that a product of each reaction $r_i \in P'$, $1 \leq i < n$, is a substrate to the subsequent reaction $r_{i+1} \in P'$. Again, we require that the production distances of the reactions added increase from sources to reaction r . This is done to restrict the potentially very large search space while still ensuring that the reactions in pathway P' can be made feasible by the reactions in these acyclic pathways. Such paths can be found by backtracking from reactions $r_i \in P'$ towards reactions with smaller production distances until a source metabolite is found.

The reactions on the augmented pathway then comprise the reaction set \mathcal{R} of the Metabolic Reconstruction Problem 1. The problem is formulated as a Problem 2 instance and solved. We obtain a result pathway that is feasible with respect to sources A . Further, the pathway is an optimal subset of the augmented pathway.

We repeat the above process of generating and augmenting an acyclic path for a fixed number of iterations, adding the metabolites on the previous result pathway to the set of source metabolites for the generation of the next pathway. Since each metabolite added to sources A is reachable from the initial sources, each successive pathway generated is also feasible. Finally, we take the union of all generated feasible pathways to be the final result.

2.5 Coding of reaction and metabolite evidence

The above method allows for use of pre-existing knowledge of metabolic reactions and metabolites. If we have evidence that a particular enzyme operates in the cell, we can set a constraint to the optimisation problem stating that the reaction has to be present in the solution. Consequently, the reconstructed model would then contain both the reaction and a good-scoring combination of reactions needed to make the network with the added reaction feasible. In the same way evidence on the existence or absence of metabolites can be encoded into the optimisation problem. Such evidence is available via metabolomics experiments, for example.

Particularly, we can take a known metabolic network as a starting point and find best-scoring reactions to add to fill the gaps in the network. This initial network could be for example derived from sequence homology evidence, or be the result of further manual curation. In this way, our method would serve as an additional tool in aiding the curator.

In practice, however, we do not set a hard constraint for reaction existence as described above. Instead, the reaction is rescored to a high value and the optimisation problem is solved. If a reaction can be feasibly connected to the network, the solver adds a good-scoring pathway which precedes the reaction to the network. On the other hand, if the reaction cannot be feasibly connected, the solver returns a solution model that does not contain the reaction. In the formulation above, the solver would simply state that the problem is unsolvable, which is not desirable.

3 Experiments

We implemented our divide-and-conquer method, denoted SCAR for Structurally Consistent Automatic Reconstruction, as a script-driven program. A Python [38] script is used to generate subproblem instances for the optimiser. The mixed integer linear programming solver `lp_solve` [3], licensed under LGPL, was used to solve the individual subproblems. The Python script merged the results from solved subproblems as the final result.

To test our method, we reconstructed a feasible metabolic network for the well-known organism *Escherichia coli* from genome data. In particular, we were interested in seeing whether the feasibility constraint that was enforced in the reconstruction would cause the method to drop metabolic reactions which were known to be present in the metabolism.

We downloaded the *E. coli* K12 protein sequences from NCBI with the accession number U00096 [4, 14]. There were 4331 coding subsequences (CDS) in total. As the enzyme database, we used UniProt version 9.3 [8] which contained 250296 protein sequences. We looked for the substring “(EC x.x.x.x)” in the textual description given for each sequence to find out which enzyme each sequence coded for. In this way, we obtained 101137 sequences with an EC number⁴.

As the reaction database \mathcal{R} , we used MetaCyc version 10.6 [15] with 6241 reactions of which 5255 had associated EC numbers. In what follows, this set of reactions is referred to as the universal metabolic network.

We compared the *E. coli* protein sequences against all protein sequences from UniProt with an EC number using BLAST [2]. E-value cutoff of BLAST was set

⁴ Each Uniprot entry specifies the enzymatic reaction only at EC number level. EC number can specify more than one concrete reaction, such as EC 2.4.1.1 which corresponds to phosphorylases operating on various sugar molecules. When the EC number of an enzyme corresponds to multiple reactions in the reaction database, each matching reaction receives the sequence homology score from that enzyme.

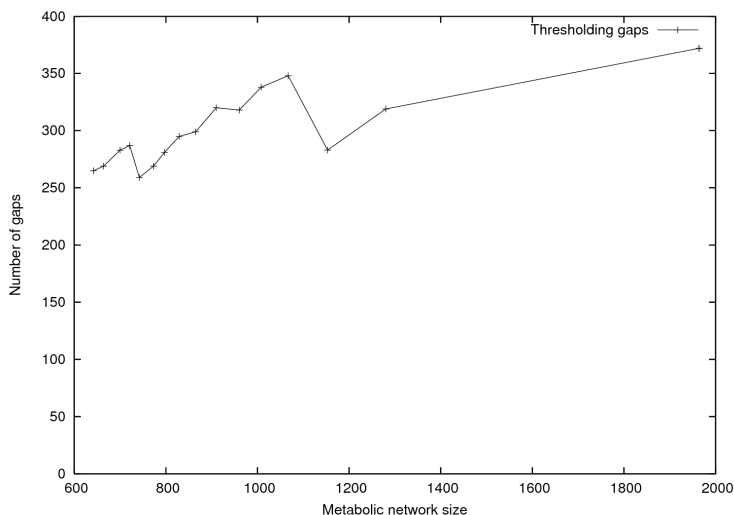


Fig. 1. Number of gaps, or reactions not connected to sources with a feasible pathway, in metabolic networks obtained with the thresholding method.

to 10 to detect remote homologs. From the BLAST scores obtained this way for each sequence pair, we derived the reaction scores for MetaCyc reactions using the equation (8).

Our method was compared against the baseline reconstruction method which is based only on sequence homology scoring. In this *thresholding method*, a reaction is chosen into the reconstruction only if the reaction score is higher than some specified threshold. Naturally, the resulting metabolic network contains gaps because the feasibility is not enforced in any way.

We reconstructed a metabolic network for *E. coli* using our method on the data discussed above. We experimented with different threshold value parameters, varying the value from 0 to 400 in increments of 25. Different values of threshold parameter model the degree of confidence we want the reconstructed model to obtain from the sequence homology: with high values of the parameter, smaller subnetworks with higher sequence homology evidence are produced, while with lower parameter values larger networks with less sequence based evidence are reconstructed. As the set of source metabolites A , we used 111 metabolites including glucose, which is the main carbon source of *E. coli*, and cofactor molecules that are known to be present in the organism in abundance, such as ATP, NAD and CO_2 .

The number of gaps in metabolic networks obtained with the thresholding method is shown in Figure 1. This value can be seen as a measure of the manual work needed to curate the initial reconstruction by hand.

A summary of the reconstruction results for both our method and the thresholding method is shown in Table 1. Our method produces smaller networks than the thresholding method as some reactions with a high score are not included in

Threshold	#Reactions (S / T)	#Gaps	#Fillers	Score (S/T)	Fill score
0	1354/1964	372	338	292.7/336.7	0
50	938/1280	319	208	410.4/498.3	7.1
100	809/1067	348	174	469.1/583.4	13.8
150	755/960	318	175	487.3/634.5	18.0
200	649/865	299	132	548.0/685.1	28.0
250	597/796	281	126	580.7/724.4	34.4
300	551/742	259	117	597.7/757.5	41.9
350	500/700	283	101	637.1/783.5	57.4
400	469/642	265	97	659.1/820.6	97.8

Table 1. Summary of results for our reconstruction method (SCAR) and the thresholding method (THRE) for the different threshold parameter values showing the numbers of reactions in results, feasibility gaps in THRE result (#Gaps), reactions with negative scores filling the gaps in the SCAR result (#Fillers), average reaction score (Score) and average score for gap filling reactions in the SCAR result (Fill score). The scores reported are $f_b(r)$ scores with $b = 0$. Only the increments of 50 are shown in the threshold value.

the result. This happens when the addition of such a reaction would also require that negative-scoring reactions were added to the network due to the feasibility constraint and the score of the resulting pathway would be negative. In this case, the pathway is not added to the network. On the average, 20–25% of reactions in our method’s result had a negative score meaning that they were used to fill the gaps in the network. Similarly, the average reaction score is smaller compared to the thresholding method as expected, because the method repairs the gaps with low-scoring reactions.

Our method took about 150 seconds on the average to run per reconstruction on a 1.6 GHz Pentium M processor. The running time can be tuned by setting the maximum number of initial paths generated. In this experiment, we used a maximum of 6000 paths which was experimentally verified to be enough for the network score $F_b(R)$ to converge.

At threshold parameter value $b = 200$, our method was able to recover 14 reactions missed by the thresholding method but found in the EcoCyc database. EcoCyc is a comprehensive, manually curated database of *E. coli* metabolism [24]. These reactions are shown in Table 2. As the score of each reaction is below the threshold value, the reason why these reactions were added to the result was to satisfy the feasibility constraint.

As an example of a gap that exists in the thresholded reconstruction but which has been filled in a SCAR reconstruction, consider the reaction phosphatidylserine decarboxylase (EC 4.1.1.65) which received a score $f_b(r) = 412$ from the BLAST alignment of *E. coli* sequence GI 1790604 and UniProt sequence P0A8K4, with $b = 200$. Our method succeeded in repairing the gap; the reaction and a feasible subnetwork that repairs the gap is shown in Figure 2.

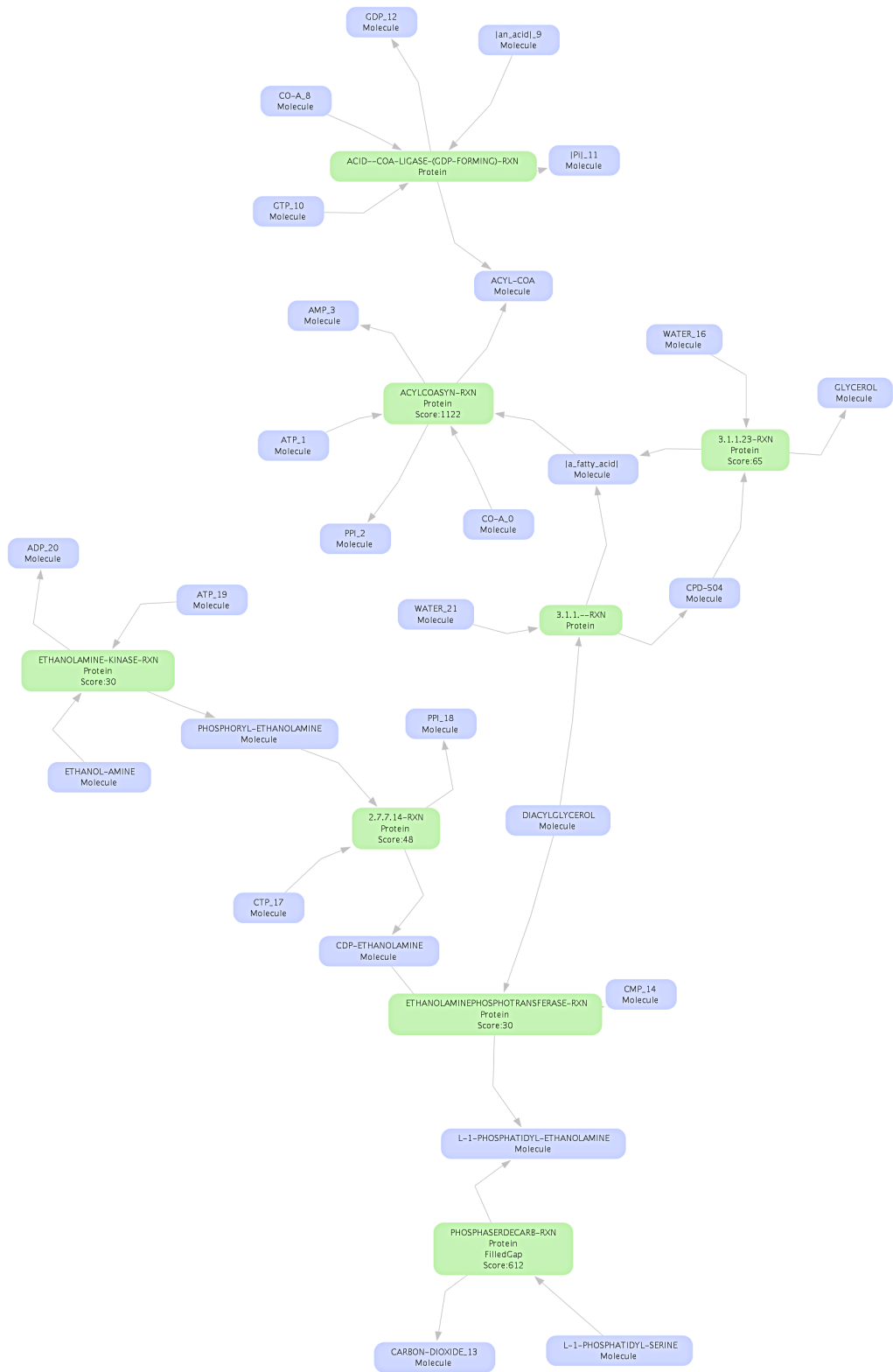


Fig. 2. Phosphatidylserine decarboxylase reaction (EC 4.1.1.65) and the network in a SCAR reconstruction that fills the gap present in the thresholding reconstruction with threshold value 200. Note that each reaction is considered to be bidirectional, thus the directionality of arrows carries no other meaning than separating substrates and products of the same reaction. The figure was generated with the **BMVis** visualisation tool [1].

Reaction	EC	Name	Score
1.1.1.283	1.1.1.283	Methylglyoxal reductase (NADPH-dependent)	37
1.13.11.16	1.13.11.16	3-carboxyethylcatechol 2,3-dioxygenase	0
BETA-PHOSPHOGLUCOMUTASE	5.4.2.6	β -phosphoglucomutase	186
CHORPYRLY	4.-.-.-	-	40
GALACTITOLPDEHYD	1.1.1.-	-	0
GLUCONOLACT	3.1.1.17	Gluconolactonase	0
NAG6PDEACET	3.5.1.25	N-acetylglucosamine-6-phosphate deacetylase	0
OXALODECARB	4.1.1.3	Oxaloacetate decarboxylase	52
PDXJ	2.6.99.2	-	0
RXN-821	-	-	0
RXN0-313	4.-.-.-	-	0
RXN0-5116	2.7.1.16	-	0
TAGAALDOL	4.1.2.40	Tagatose-bisphosphate aldolase	69
TAGAKIN	2.7.1.144	Tagatose-6-phosphate kinase	124

Table 2. Reactions included in the EcoCyc database that were not found by the thresholding method (threshold value 200) because of low reaction scores, but which were included in the SCAR reconstruction. MetaCyc reaction identifiers without the trailing _RXN shown. A dash signifies a missing EC number or name in MetaCyc.

4 Discussion

In this article we introduce a computational method to assist the reconstruction of a metabolic network of an organism based on its genome information. The method is based on optimization techniques and graph traversal algorithms. As a distinctive feature, the presented method combines the search of reactions that are most likely catalyzed by the genes of the target organism and the filling the gaps in the reconstructed metabolic network to a single computational step. It is easy to incorporate experimental evidence, such as information about experimentally observed metabolites or reactions with the method to improve the quality of the reconstructed metabolic network models.

As the present method constructs gapless metabolic networks, the reconstructed models can contain also reactions that have no known catalyzing enzyme, as long as these reactions improve the feasibility of the model. The advantages of this feature are twofold. First, we can augment the reaction database utilized by the reconstruction algorithm with computer-generated, or hypothetical, reactions [20] without the knowledge of enzymes possibly catalyzing these reactions. Second, the identification of an unannotated but structurally necessary reaction can serve as a hint for finding the gene responsible for that particular function. This capability is beyond most current computational methods for metabolic reconstruction.

The present method shares some properties with recently introduced method called GapFill [28]. The main difference between GapFill and the present method is found from the weaker definition of the reaction gap applied in the GapFill. In GapFill, reaction gaps are thought to be removed as soon as the network produces substrate metabolites for each of its internal reactions – even if the

network is unable to produce these metabolites from its external substrates. This formulation easily leads to situations where the gaps in a draft network are filled by small cycles where reactions produce substrates for each others, but that are disconnected from the rest of the network. In the present method, on the other hand, we require the substrates of each reaction in the network have to be produced from the external sources. We argue that our stronger definition of the reaction gaps is biologically more relevant: the network model where some metabolites cannot be produced from the external sources is clearly incomplete. For example in the most tasks of constraint based modelling of metabolism, the reactions whose substrates cannot be produced from the external substrates are irrelevant, and can be removed from the model as a preprocessing step. Furthermore, in GapFill the gaps in the network are filled locally, and only the number of reactions needed to fill the gaps is considered in the optimization. The present method, on the other hand, looks for globally optimal modifications to simultaneously fill all the gaps, taking also the available genomic evidence for the existence of gap-filling reactions in the target organism into account.

To conclude, we believe that the present method is able to produce useful suggestions about the structure of a metabolic network to guide a domain expert in a very time-consuming task of metabolic reconstruction. To further improve the accuracy of the reconstructed metabolic network models, we will investigate alternative ways of scoring the reactions in a database. These alternatives contain more advanced methods for detecting homologous sequences between the enzymes in a database and the genome of the target organism that might share a function, as well as the inclusion of the other type of data, such as whole-cell metabolome measurements, into the computation of reaction scores.

5 Acknowledgments

We would like to thank Antti Tani for the initial computational experiments, Taneli Mielikäinen and Paula Jouhten for valuable discussions, the BioMine project at the Department of Computer Science, University of Helsinki for the BMVis network visualisation tool, and the reviewers for their comments that helped to improve the manuscript.

References

1. BMVis graph visualisation tool. <http://www.cs.helsinki.fi/group/biomine>. Department of Computer Science, University of Helsinki.
2. S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.
3. M. Berkelaar, K. Eikland, and P. Notebaert. lp_solve: Open source (mixed-integer) linear programming system, 2005. Multi-platform, pure ANSI C / POSIX source code, Lex/Yacc based parsing. Version 5.1.0.0 dated 1 May 2004. GNU LGPL (Lesser General Public Licence). http://groups.yahoo.com/group/lp_solve.

4. F. R. Blattner, G. Plunkett 3rd, C. A. Bloch, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mayhew, J. Gregor, N. W. Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau, and Y. Shao. The complete genome sequence of *Escherichia coli* K-12. *Science*, 277:1453–1474, 1997.
5. A. P. Burgard, E. V. Nikolaev, C. H. Schilling, and C. D. Maranas. Flux coupling analysis of genome-scale metabolic network reconstructions. *Genome Research*, 14(2):301–312, 2004.
6. R. Caspi, H. Foerster, C. A. Fulcher, R. Hopkinson, J. Ingraham, P. Kaipa, M. Krummenacker, S. Paley, J. Pick, S. Y. Rhee, C. Tissier, P. Zhang, and P. D. Karp. Metacyc: A multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, 34:D511–D516, 2006.
7. L. Chen and D. Vitkup. Predicting genes for orphan metabolic activities using phylogenetic profiles. *Genome Biology*, 7(R17):1777–1782, 2006.
8. The UniProt Consortium. The universal protein resource (UniProt). *Nucleic Acids Research*, 35:D193–197, 2007.
9. S. J. Cordwell. Microbial genomes and "missing" enzymes: redefining biochemical pathways. *Arch Microbiol*, 172:269–279, 1999.
10. M. Csete and J. Doyle. Bow ties, metabolism and disease. *TRENDS in Biotechnology*, 22(9):446–450, 2004.
11. N. Duarte, S. Becker, N. Jamshidi, I. Thiele, M. Mo, T. Vo, R. Srivas, and B. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proceedings of National Academy of Science, USA*, 104(6):1777–1782, 2007.
12. N. Duarte, Herrgård M., and B. Palsson. Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Research*, 14(7):1298–1309, 2004.
13. J. Edwards and B. Palsson. The *Escherichia coli* MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. *Proceedings of National Academy of Science, USA*, 97(10):5528–5533, 2000.
14. M. Riley *et al.* *Escherichia coli* K-12: a cooperatively developed annotation snapshot. *Nucleic Acids Res.*, 34(1):1–9, 2006.
15. R. Caspi *et al.* The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 36(Database issue):D623–D631, 2008.
16. J. Förster, I. Famili, P. Fu, B. Palsson, and J. Nielsen. Genome-scale reconstruction of the *Saccharomyces cerevisiae* metabolic network. *Genome Research*, 13(2):244–253, 2003.
17. C. Francke, R. J. Siezen, and B. Teusink. Reconstructing the metabolic network of a bacterium from its genome. *Trends Microbiol*, 13(11):550–558, 2005.
18. M. Galperin and E. Koonin. Who's your neighbor? new computational approaches for functional genomics. *Nat Biotechnol*, 18:609–613, 2000.
19. M. Green and P. D. Karp. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC Bioinformatics*, 5(76), 2004.
20. V. Hatzimanikatis, C. Li, J. A. Ionita, C. S. Henry, M. D. Jankowski, and L. J. Broadbelt. Exploring the diversity of complex metabolic networks. *Bioinformatics*, 21(8), 2005.
21. M. Herrgård, S. Fong, and B. Palsson. Identification of genome-scale metabolic network models using experimentally measured flux profiles. *PLoS Computational Biology*, 2(7):e72, 2006.
22. IUBMB. *Enzyme Nomenclature*. Academic Press, 1992.

23. P. Karp, S. Paley, and P. Romero. The pathway tools software. *Bioinformatics*, 18:S225–S232, 2002.
24. I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil, and P. D. Karp. EcoCyc: A comprehensive database resource for *Escherichia coli*. *Nucleic Acids Research*, 33:D334–D337, 2005.
25. P. Kharchenko, L. Chen, Y. Freund, D. Vitkup, and G. M. Church. Identifying metabolic enzymes with multiple types of association evidence. *BMC Bioinformatics*, 7(177), 2006.
26. P. Kharchenko, G. M. Church, and D. Vitkup. Filling gaps in a metabolic network using expression information. *Bioinformatics*, 20:I178–I185, 2004.
27. P.-J. Kim, D.-Y. Lee, T. Kim, K. Lee, H. Jeong, S. Lee, , and S. Park. Metabolite essentiality elucidates robustness of *Escherichia coli* metabolism. *Proceedings of National Academy of Science, USA*, 104(34):13638–13642, 2007.
28. V. Kumar, M. Dasika, and C. Maranas. Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics*, 8(212):13638–13642, 2007.
29. E. Marcotte. Computational genetics: finding protein function by nonhomology methods. *Curr Opin Struct Biol*, 10:359–365, 2000.
30. A. McGuire, J. Hughes, and G. Church. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Res*, 10:744–757, 2000.
31. A. Osterman and R. Overbeek. Missing genes in metabolic pathways: a comparative genomics approach. *Current Opinion in Chemical Biology*, 7:238–251, 2003.
32. P. Pharkya and C. D. Maranas. An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metabolic Engineering*, 8(1):1–13, 2006.
33. E. Pitkänen, A. Rantanen, J. Rousu, and E. Ukkonen. 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, 2005.
34. O. Resendis-Antonio, J. Reed, S. Encarnación, J. Collado-Vides, and B. Palsson. Metabolic reconstruction and modeling of nitrogen fixation in *Rhizobium etli*. *PLoS Computational Biology*, 3(10):13638–13642, 2007.
35. C. H. Schilling, D. Letscher, and B. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, 203(3):228–248, 2003.
36. S. Schuster, D. A. Fell, and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic network. *Nature Biotechnology*, 18:326–332, March 2000.
37. J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles. Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420:190–193, 2002.
38. G. van Rossum and F. L. Drake, Jr. *An Introduction to Python*. Network Theory Ltd, 2006.
39. Y. Wolf, I. Rogozin, A. Kondrashov, and E. Koonin. Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Res*, 11:356–372, 2001.