Supplementary material

The search engine of MetaRoute is an extended version of a graph-based approach which we recently reported [Blum and Kohlbacher, 2008]. Several novel concepts (explained in the following two sections) improve both the search time and quality of the result. The search performance of MetaRoute, compared to our previous work, is presented in the last section.

Graph representation

Our graph representation integrates precalculated atom mapping rules. Therefore, each reaction in the network is decomposed into a set of all possible educt/product pairs where at least one atom is transferred from the educt to the product according to the atom mapping rule of that reaction. Then each node in the graph represents a unique educt/product atom mapping pair (E_i, P_j). All reactions which have such a pair in common are associated with the corresponding node. This allows a more compact representation of the reactions because frequent metabolic transitions like acetyl-CoA/CoA or glutamate/2-oxoglutarate, which are shared by 117 and 67 reactions respectively in the 'super network', are summarized by one single node. The reverse reactions are represented by product/educt pair nodes. Each edge in the graph connects two nodes (E_i, P_j) and (E_k, P_l) if $P_j = E_k$ and if at least one atom is transferred from E_i to P_l according to the sequential application of the atom mapping rules of the two nodes.

Although this (transition) graph requires many more nodes and edges compared with other graph representations, it is more suited to our path finding algorithm. Each biochemically feasible route consisting of two successive reaction steps is implicitly represented by one edge in the transition graph because each node codes one educt/product conversion. Therefore, the path finding algorithm does not have to deal with a huge number of routes that contain irrelevant metabolic conversions like glucose-6-phosphate $\rightarrow$ ATP $\rightarrow$ AMP because there will be no edge between the nodes representing the educt/product pairs glucose-6-phosphate, ATP and ATP, AMP.

The computational complexity can be further reduced by the restriction of only tracing the transition of one atom type. For example, if we are interested in questions concerning the carbon metabolism, we can simply ignore nodes and edges where no carbon atom is transferred. The same can be done for the nitrogen, sulfur, or phosphorous metabolism.

In the weighted metabolic networks approach [Croes *et al.*, 2005] a bipartite graph is used where the edges connect the compound nodes with the reaction nodes. Each (compound) node is assigned a weight equal to its degree (number of reactions it is participating in as educt or product). Then the lightest path search significantly reduces the probability of finding irrelevant routes containing network hubs or pool metabolites (e.g. ATP, H_2O)

as main intermediates. However, this approach fails to find routes passing pathways of the core metabolism (like glycolysis or TCA cycle) because frequently occuring compounds (like pyruvate or acetyl-coA) have to be traced. Therefore, we create an additional weight that also considers the context of the traced reactions as a counterpart to the compound weight. The context of a reaction contains all compounds of that reaction which are not used as intermediates in the path search. Each reaction $R_{k,(i,j)}$ associated with the transition node (E_i, P_j) gets a context weight for that node. The context consists of all educts $E_k, k \neq i$ and products $P_k, k \neq j$ of $R_{k,(i,j)}$. The weight of $R_{k,(i,j)}$ is the summed context weight of its context compounds. The higher (lower) the number of reactions in the metabolic 'super network' a compound participates in, the lower (higher) is its context weight. We use a function (based on piecewise linear interpolation) that maps the reaction count to the context weight and is defined as follows:

$$cw(i) = \begin{cases} 1 & \text{if } rc(i) > b_1 \\ 1 + \frac{(rc(i)-b_1)(b_3-1)}{b_2-b_1} & \text{if } b_2 \leq rc(i) \leq b_1 \\ b_3 + \frac{(rc(i)-b_2)(b_2-b_3)}{b_3-b_2} & \text{if } b_3 \leq rc(i) < b_2 \\ b_2 + \frac{(rc(i)-b_3)(b_1-b_2)}{1-b_3} & \text{if } 1 \leq rc(i) < b_3 \end{cases}$$

where:

- $cw(i)$: the context weight of compound i in the network
- $rc(i)$: the reaction count of compound i (number of reactions it is participating in as educt or product)
- b_1, b_2, b_3: empirical bounds which seperate compounds in the network in frequent occuring compounds which will obtain low context weights, in compounds of medium frequency and in rare occuring compounds which will get high context weights ($b_1 = 500$, $b_2 = 100$, $b_3 = 10$ in case of the KEGG 'super network')

For example, the reactions EC 1.4.1.4 and EC 2.6.1.42 share the transformation of 2-oxoglutarate to glutamate where EC 1.4.1.4 uses NADPH, $NADP^+$, NH_3 and H_2O as co-substrates contrary to EC 2.6.1.42 using valine and 2-keto-isovalerate. The exclusive presence of pool metabolites in the context (all co-substrates) of EC 1.4.1.4 results in the lower weight of eight compared to the weight of 239 for the context of EC 2.6.1.42. This weight makes sure that a biosynthesis route of glutamate via 2-oxoglutarate prefers to trace EC 1.4.1.4 instead of EC 2.6.1.42 which requires the production and consumption of further amino acids. The compound and reaction weights are incorporated into our transition graph where each edge represents the intermediary metabolite I_m of two succeeding educt/product transition nodes (E_i, I_m) and (I_m, P_j). The weight of each edge is assigned the number of reactions (in the network) in which I_m participates plus the minimum context

weight of the reactions $R_{k,(m,j)}$ associated with the target transition node (I_m, P_j).

Note that a path in the graph can code multiple metabolic routes if more than one reaction is associated with at least one node in the path. The combined weighting is more suited to routes passing the core metabolism like glycolysis and the TCA cycle or routes of the purine biosynthesis which involve 'hub compounds' like ADP as main intermediates.

Path finding algorithm

We use Eppstein's k-shortest path algorithm [Eppstein, 1998] which efficiently computes the first k-shortest (or lightest) paths between two given nodes in a directed graph. Therefore, we create in each search a start node s and end node e representing the source metabolite E_s and product P_e where s is connected to all nodes (E_s, P_j) and e to (E_i, P_e). The weights of the edges connecting s and e with the graph are calculated as described above. Furthermore, the algorithm is modified to consider the atom mapping rules. Each extracted path is validated by the sequential application of the atom mapping rules. Starting with the first node (educt/product conversion (E_s, P_1)) the atoms of E_s are mapped to P_1 according to the associated atom mapping rule. Then for each subsequent educt/product transition (E_i, P_j), only those atoms of E_i reached by the mapping in the preceding transition step are mapped to P_j. The path is rejected as irrelevant if no atom of the source is contained in the product.

Search prerformance

The search performance of MetaRoute was evaluated using the same experimental setting as described in our previous work [Blum and Kohlbacher, 2008]. This enables to demonstrate how the performance is influenced by the novel concepts of MetaRoute. A similar setting was proposed by Croes *et al.* [2005] for evaluating the search performance of graph-based approaches.

Experimental setting

The search performance is evaluated by trying to find experimental verified biotransformation routes in the metabolic network of *E. coli* at genome scale. For this purpose, all annotated routes of the small molecule metabolism with at least three reactions were extracted from EcoCyc (137 overall). Given the main source and target metabolites of the annotated routes as start and end nodes, we calculated the lightest path constrained to use the first as well as the last reaction of the annotated route. If n annotated routes share the same main source as well as target metabolites and start as well as end reaction, we computed the n lightest paths. The quality of the routes found was measured by comparing the intermediate compounds and reactions with the

annotated routes, and is expressed using sensitivity, specificity and relevance score, which are defined as follows:

$$sensitivity = \frac{tp}{tp+fn}$$
$$specificity = \frac{tp}{tp+fp}$$
$$relevance = \frac{sensitivity+specificity}{2} * smc$$

where:

- tp (true positives): The number of compounds and reactions of the route found which are also present in the annotated route. The first and last compounds and reactions are not considered.
- fp (false positives): The number of compounds and reactions of the route found which are not present in the annotated route.
- fn (false negatives): The number of compounds and reactions of the annotated route which are not present in the route found.
- smc (structural moiety constraint): This value is set to 1 if the route found fulfills the structural moiety constraint, and set to 0 otherwise.

If an extracted route was not identical to an annotated one and contains reactions for which no atom mapping rules could be calculated [Blum and Kohlbacher, 2008], we manually checked the structural moiety constraint. Note, that this evaluation precedure produces only relative performance measures useful for comparing different search strategies because novel routes could be very different to the annotated ones. More details are available elsewhere [Blum and Kohlbacher, 2008].

In case of MetaRoute we always use the carbon network (with carbon as the only traceable atom type) except for the sulfate reduction pathway (EcoCyc ID: SO4ASSIM-PWY) because sulfate and hydrogen sulfide are used as source and target metabolites. Here, we used the sulfur network (with sulfur as the only traceable atom type). Furthermore, the novel combined weighting scheme is used. Because the network under study is approx. five times smaller than the KEGG 'super network', we use five times smaller frequency borders (100, 20, 2).

Results

The search results are shown in Tab. 1. Compared to our previous work, MetaRoute shows significant improvements with respect to all performance measures. The amount of the increase is approximately eight per cent. Now, nearly all of the extracted routes fulfill the structural moiety constraint (99%). Especially the routes of the core metabolism (glycolysis and TCA cycle) and the routes of the purine biosynthesis are better predicted.

Table 1: The search results for 137 experimentally verified biotransformation routes extracted from EcoCyc are shown here. The results of the verified routes present only in glycolysis, the TCA cycle and the purine biosynthesis are also shown. In each section, the first row represents the search approach of our previuos work and the second that of MetaRoute including all of the novel concepts. The columns show the average sensitivity (sens), specificity (spec), structural moiety constraint (smc) and relevance score (rel).

experiment	approach	sens	spec	smc	rel
all routes	**Blum and Kohlbacher [2008]**	0.86	0.87	0.91	0.86
	MetaRoute	0.93	0.95	0.99	0.94
glycolysis	**Blum and Kohlbacher [2008]**	0.73	0.80	1.00	0.77
	MetaRoute	0.96	0.96	1.00	0.96
TCA cycle	**Blum and Kohlbacher [2008]**	0.46	0.67	1.00	0.56
	MetaRoute	1.00	1.00	1.00	1.00
purine syn.	**Blum and Kohlbacher [2008]**	0.67	0.70	0.75	0.69
	MetaRoute	0.93	0.94	1.00	0.94

References

Blum,T., Kohlbacher,O. (2008) Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks, *J. Comput. Biol.*, **15(6)**.

Croes,D., Couche,F., Wodak,S.J., van Helden,J. (2006) Metabolic PathFinding: inferring relevant pathways in biochemical networks, *Nucl. Acids Res.*, **33**, W326-W330.

Eppstein,D. (1998) Finding the k Shortest Paths, *SIAM Journal on Computing*, **28**, 652-673.