Cover Letter

Paper Title:
Mining metabolic networks for optimal drug targets

Contact Author:
Name: Tamer Kahveci
Email: tamer@cise.ufl.edu

Authors:
Name: Padmavati Sridhar, Bin Song, Tamer Kahveci and Sanjay Ranka
Address:
Computer and Information Science and Engineering,
University of Florida,
Gainesville, FL, 32611
E-mail: {psridhar, bsong, tamer, ranka}@cise.ufl.edu

The specific PSB session that should review the paper or abstract:
Molecular Bioinformatics for Diseases: Protein Interactions and Phenomics

The submitted paper contains original, unpublished results, and is not currently under consideration elsewhere. All co-authors concur with the contents of the paper.

.

MINING METABOLIC NETWORKS FOR OPTIMAL DRUG TARGETS [*]

PADMAVATI SRIDHAR, BIN SONG, TAMER KAHVECI[†]AND SANJAY RANKA

Computer and Information Science and Engineering,
University of Florida,
Gainesville, FL, 32611
E-mail: {psridhar, bsong, tamer, ranka}@cise.ufl.edu

Recent advances in bioinformatics promote drug-design methods that aim to reduce side-effects. Efficient computational methods are required to identify the optimal enzyme-combination (i.e., drug targets) whose inhibition, will achieve the required effect of eliminating a given target set of compounds, while incurring minimal side-effects. We formulate the optimal enzyme-combination identification problem as an optimization problem on metabolic networks. We define a graph based computational damage model that encapsulates the impact of enzymes onto compounds in metabolic networks. We develop a branch-and-bound algorithm, named *OPMET*, to explore the search space dynamically. We also develop two filtering strategies to prune the search space while still guaranteeing an optimal solution. They compute an upper bound to the number of target compounds eliminated and a lower bound to the side-effect respectively. Our experiments on the human metabolic network demonstrate that the proposed algorithm can accurately identify the target enzymes for known successful drugs in the literature. Our experiments also show that OPMET can reduce the total search time by several orders of magnitude as compared to the exhaustive search.

1. Introduction

In pharmaceutics, the development of every drug mainly involves target identification, validation and lead inhibitor identification [9]. Traditional drug discovery approaches focus more on the efficacy of drugs than their toxicity (untoward side effects). Lack of predictive models that account for the complexity of the inter-relationships between the metabolic processes often leads to drug development failures. Toxicity and/or lack of efficacy can result if metabolic network components other than the intended target are affected. The current focus is on identification of biological targets (gene products, such as enzyme or protein) for drugs, which can be manipulated to produce the desired effect (of curing a disease) with minimum disruptive side-effects [23,27].

Enzymes catalyze reactions, which produce metabolites (compounds) in the metabolic networks of organisms. Enzyme malfunctions can result in the accumulation of certain compounds which may result in diseases. We term such compounds

[*] this work is supported in part by the national science foundation under grant itr 0325459 and dbi - 0606607. any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the national science foundation.

[†] to whom correspondence should be addressed

as *Target Compounds* and the remaining compounds as *Non-Target compounds*. For instance, the malfunction of enzyme *phenylalanine hydroxylase* causes buildup of the amino acid, phenylalanine, resulting in phenylketonuria [26], a disease that causes mental retardation. It is, therefore, needed to locate the optimal enzyme set which can be manipulated by drugs to prevent the excess production of target compounds with minimal damage. Formally, we define *damage* of inhibiting an enzyme (or a set of enzymes) as the number of non-target compounds whose production is stopped by the inhibition of that enzyme (or set of enzymes).

Given a metabolic network and a set of target compounds, we consider the problem of identifying the set of enzymes whose inhibition eliminates the target compounds and incurs minimum damage. Evaluating all enzyme combinations is not feasible as the number of such combinations increases exponentially with the number of enzymes. Hence, more efficient computational methods are needed. In our earlier work [25], we developed a heuristic solution to this problem. Here, we propose *OPMET*, an **Op**timal enzyme drug target identification algorithm based on **Met**abolic networks, to solve this problem optimally. This paper has two main contributions. 1) We propose a branch-and-bound algorithm, named *OPMET*, to explore the search space. Based on the damage model, OPMET dynamically updates the priorities as the search space is explored. 2) We develop two filtering approaches which are combined with the OPMET to prune the search space while still guaranteeing an optimal solution.

Our experiments on the human metabolic network demonstrates that the proposed algorithm can accurately identify the target enzymes for known successful drugs in the literature. Our experiments also show that our methods reduce the total search time by several orders of magnitude as compared to the exhaustive search. OPMET prunes 91.6 % of the search space. It generates the optimal enzyme combination within the exploration 0.005 % of the search space on average.

The rest of the paper is organized as follows. Section 2 formally defines the problem and describes our proposed cost model. Section 3 presents the proposed OPMET algorithm with filtering strategies. Section 4, discusses experimental results. Section 5 discusses the related work. Section 6 concludes the paper.

2. Problem definition

We develop a graph based representation that captures the interactions between reactions, compounds, and enzymes. Our graph representation is a variation of the boolean network model [24,16]. R, C, and E denote the set of reactions, compounds, and enzymes respectively. The vertex set consists of all the members of $R \cup C \cup E$. A vertex is labeled as reaction, compound, or enzyme based on the entity it refers to. Let V_R, V_C, and V_E denote the set of vertices from R, C, and E. A directed edge from vertex x to vertex y is then drawn if one of the following three conditions holds: (1) x represents an enzyme that catalyzes the reaction represented by y. (2) x corresponds to a substrate for the reaction represented by y. (3) x represents a reaction that produces the compound mapped to y.

Figure 1 illustrates a small hypothetical metabolic network. In this figure, C_4 is the target compound (i.e., the production of C_4 should be stopped). In order to stop the production of C_4, R_2 has to be prevented from taking place. The obvious solution is to disrupt one of its catalyzing enzymes (E_2 in this case). Another is by stopping the production of one of its reactant compounds (C_2 or C_3 in this case). If we stop the production of C_2, we need to recursively look for the enzyme which is indirectly responsible for its production (E_1 in this case). Thus, the production of the target compound can be stopped by manipulating either E_1 or E_2.

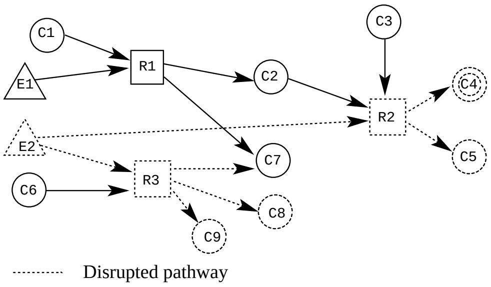

Figure 1. A graph constructed for a hypothetical metabolic network with three reactions R_1, R_2, and R_3, two enzymes E_1 and E_2, and nine compounds $C_1, \cdots, C_9$. Circles, rectangles, and triangles denote compounds, reactions, and enzymes respectively. Here, C_4 (shown by double circle) is the target compound. Dotted lines indicate the subgraph removed due to inhibition of enzyme E_2.

Figure 1 shows the disruption of E_2 and its effect on the network. Inhibiting E_2 results in the knock out of compounds C_5, C_8 and C_9 in addition to the target compound, C_4. Note that the production of C_7 is not stopped since it is produced by R_1 even after the inhibition of E_2. We define the number of non-target compounds knocked out as the *damage*, the manipulation of an enzyme set causes to the metabolic network. In this case, the damage of inhibiting E_2 is 3 (i.e., C_5, C_8 and C_9). The damage of inhibition of E_1 is 2 (i.e., C_2 and C_5). The important observation is that E_1 and E_2 both achieve the effect of disrupting the target compound, C_4. Hence, E_1 and E_2 are both potential drug targets. However, E_1 is a better drug-target than E_2 since it causes lesser damage.

Formally the optimal enzyme combination identification problem is: "*Given a set of target compounds T ($T \subset C$), find the set of enzymes X ($X \subseteq E$) with minimum damage, whose inhibition stops the production of all the compounds in T.*"

For simplicity, we assume that the input compounds to all reactions are present in the network and that there are no external inputs. Different enzymes and compounds may have varying levels of importance in the metabolic network. We consider all the enzymes and compounds to be of equal importance. This assumption can be relaxed by assigning weights to enzymes and compounds based on their role in the network. Also, we are not incorporating back-up enzyme activities [20] in this paper. This can be achieved by creating vertices for sets of enzymes in our graph representation. However, we do not discuss these extensions in this paper.

3. Proposed methods

This section proposes OPMET, a branch and bound algorithm that considerably reduces the number of possible combinations to be searched while still guaranteeing to find an optimal solution. Section 3.1 describes the basic branch and bound algorithm. Our prioritization (Section 3.2) and filtering (Section 3.3) strategies further improve this algorithm by reducing the search space.

3.1. *State space and basic search strategy of OPMET*

Let $E = \{E_i| \forall i,\ 1 \leq i \leq m\}$ denote the set of enzymes for a metabolic network. The search space is modeled as a tree structure. Every node of this tree corresponds to a state in the search space and it is represented by a 4-tuple $([e_{\pi_1}, e_{\pi_2}, \cdots, e_{\pi_m}], k, d, remove)$. Here, $\pi_1, \cdots, \pi_m$ is a permutation of 1, 2, $\cdots$, m. The first parameter corresponds to the state of all the enzymes (i.e., e_{π_i} corresponds to enzyme E_i). $e_{\pi_i} = 1$ if E_i is inhibited. Otherwise, $e_{\pi_i} = 0$. The parameter k indicates that the first k enzymes are considered at that search state. The decision to inhibit or not inhibit has been fixed for enzymes from 1 to $k - 1$ and we now set $e_{\pi_k} = 1$ and $e_{\pi_i} = 0$, $\forall i$, $k < i \leq m$. The damage incurred due to inhibited enzymes at that state is represented by d. The final parameter, $remove$, is a boolean variable. It takes value *True* if the inhibited enzymes stop the production of all the target compounds. Otherwise, it is set to *False*. We call a node with $remove$ = True as a *true node*, and a node with $remove$ = False as a *false node*.

OPMET Algorithm: We start with the *root node* ([0, 0, $\cdots$, 0], 0, 0, False) indicating that all enzymes are present in the network. As the search space is traversed, we keep the true node with the minimum damage found so far as the *current true solution* and store the associated damage value as D, the *global cut-off threshold*. D is initialized to the number of compounds in the network. At any point, we have an *active set* of nodes A, stored in a stack structure. A contains the nodes currently being considered. Let node $N = ([e_{\pi_1}, e_{\pi_2}, \cdots, e_{\pi_m}], k, d, remove)$ be the node on top of this stack (i.e., the node to be evaluated). There are three cases:

• *Case 1:* N has damage $d > D$. In this case we prune the subtree rooted at N. We then backtrack.

• *Case 2:* N is a true node with damage $d < D$. In this case, we save N as the *current true solution* and update D with the damage value of N. We then backtrack.

• *Case 3:* N is a false node with damage $d < D$. In this case, we insert N in the active set A for backtracking purposes. We then create a new node N' by setting $e_{\pi_{k+1}} = 1$ in N (i.e., we inhibit the enzyme $E_{\pi_{k+1}}$). The resulting node is $N' = ([e_{\pi_1}, e_{\pi_2}, \cdots, e_{\pi_m}], k+1, d', remove')$. The node N' is evaluated in the next step similarly.

Backtracking involves following steps. First we pick the top node from the active nodes stack A. Let $N' = ([e'_{\pi_1}, e'_{\pi_2}, \cdots, e'_{\pi_m}], k', d', remove')$ denote this node. We then set $e_{\pi_{k'+1}} = 0$ (indicating the node we are backtracking from) and $e_{\pi_{k'+2}} = 1$ in N' (i.e., we inhibit the enzyme $e_{\pi_{k'+2}}$). The resulting node becomes the node to be evaluated in the next step. The first two cases above stop expanding the tree at the current node. The former one implies that the current node is a possible solution. The latter one implies that the current node incurs too much damage to lead to a possible solution. The third case happens when the current node does not stop production of all the target compounds, but the damage is lower than the damage of the current best solution. Such nodes may produce a possible solution with the inhibition of more enzymes. Thus, they need to be explored further to ensure that we find an optimal solution. The search terminates when there are no more nodes to explore. At this stage, the current true solution is the *optimal solution*.

3.2. *Improving the OPMET algorithm by enzyme prioritization*

In order to benefit from the pruning power of OPMET cases 1 and 2 (see Section 3.1), we need to compute the permutation $\pi_1, \cdots, \pi_m$ carefully. The earlier we place the enzymes in the optimal solution in this permutation, the better, as OPMET reaches the optimal solution earlier under such an ordering. Thus, reaching the solution with the smallest possible damage value (i.e., the *optimal solution*) increases the chances of pruning the remaining nodes of the search tree. This section develops a cost model to prioritize the enzymes dynamically.

3.2.1. *Cost model*

We develop a cost model as the basis for enzyme ordering in OPMET. This cost model takes both the observed and potential damage resulting from the inhibition of an enzyme set into the cost computation. For each enzyme $E_i \in E$, we compute a weight $W(E_i)$ as $W(E_i) = 0$ if E_i inhibited and $W(E_i) = 1$ otherwise. We assign fractional weights between 0 and 1 to the reaction and compound nodes and the edges. Intuitively, the weight of a node or edge denotes the rate at which that node or edge appears in the network. The weight of each node is calculated as follows:

• **Cost Rule 1:** Let R_j be a reaction node. Let w_i, $1 \leq i \leq k$, denote the weights of the incoming edges to R_j. We compute the weight of R_j as $W(R_j) = \min_{i=1}^{k}\{w_i\}$. This is intuitive since a reaction takes place only if all the inputs are present.

• **Cost Rule 2:** Let C_j be a compound node. Let w_i, $1 \leq i \leq k$, denote the weights of the incoming edges to C_j. We compute the weight of C_j as $W(C_j) = \frac{1}{k}\sum_{i=1}^{k}\{w_i\}$. This weight evaluates to zero only if all the reactions that produce C_j stops.

We define the weight of an edge as the weight of the node for which it is the outgoing edge. In order to compute the cost of E_i, we set the weight of E_i to zero (i.e., $W(E_i) = 0$). The weights of all the reaction and compound nodes are assigned progressively by a breadth-first search, according to the above scheme. The weights of all the nodes and edges which can be reached from E_i are recomputed to reflect the change. We define an *impact vector* for each enzyme based on the effects of its inhibition.

Definition 3.1. Given a network with n compounds, C_j, $1 \leq j \leq n$. Let $W_i(C_j)$ denote the weight of the node corresponding to C_j after the inhibition of enzyme E_i. We define the *impact vector* of E_i as $I(E_i) = [W_i(C_1), W_i(C_2), \cdots, W_i(C_n)]$. We term $W_i(C_j)$ as the *impact of* E_i *on* C_j, $\forall j$. ■

The impact vector of an enzyme approximates the amount of each compound that remains after the inhibition of that enzyme. Every entry of the impact vector is a fractional number between 0 and 1, where 0 indicates that the corresponding compound does not exist after inhibition of the corresponding enzyme. We define the cost of an enzyme as follows:

Definition 3.2. Given a network with n compounds, C_j, $1 \leq j \leq n$. Assume that the compounds $C_j, \forall j, 1 \leq j \leq k \leq n$ constitute the set of target compounds. Assume that the remaining compounds $C_j, \forall j, k+1 \leq j \leq n$ constitute the non-target compounds. Let $I(E_i) = [W_i(C_1), \cdots, W_i(C_n)]$ denote the impact vector of E_i. We define the *cost* of E_i as cost$(E_i) = I(E_i) \cdot V^T$, where $V = [v_1, \cdots, v_n]$ is the *normalization vector*: $v_i = \frac{n-k}{k}$ for $1 \leq i \leq k$, and $v_i = -(\frac{n-k}{n-k})$ for $k < i \leq n$. ■

Each target compound contributes a positive value and each non-target compound contributes a negative value to the cost of an enzyme. This is justified since the cost promotes removal of target compounds and demotes the removal of non-target compounds.

3.2.2. ***Ordering of enzymes in OPMET***

Based on the impact vector of individual enzymes, we propose an incremental strategy for ordering of enzymes in OPMET. Let $R = [r_1, r_2, \cdots, r_n]$ denote the remaining fractions of compounds. Here, $r_i \in [0, 1]$ corresponds to compound C_i, $\forall i$. We initialize $r_i = 1, \forall i$ indicating that all compounds are being produced without any disruption. Let V be the normalization vector as given in Definition 3.2. Let $I(E_i)$ be the impact vector of enzyme E_i (see Definition 3.1). Assume $N = ([e_{\pi_1}, e_{\pi_2}, \cdots, e_{\pi_m}], k, d, remove)$ be the node currently being evaluated (i.e., the decision to inhibit or not inhibit has been fixed for $e_{\pi_1}, e_{\pi_2}, \cdots, e_{\pi_{k-1}}$). We now need to decide which enzyme has to be evaluated next. In details, for every enzyme in the remaining enzyme set (e_{π_i}, $\forall i, k \leq i \leq m$), we compute the new remaining fractions of compounds (R_i). This is done by a Vector Direct Product of R and the impact vector of E_i ($I(E_i)$). That is, $R_i = R \odot I(E_i)$. Vector direct product is defined as $X \odot Y = [x_1 y_1, x_2 y_2, \cdots, x_n y_n]$, where $X = [x_1, \cdots, x_n]$ and $Y = [y_1, \cdots, y_n]$. The resulting vector R_i is an approximation to the impact of inhibition of the enzyme E_i in addition to already inhibited enzymes. This is justified since the quantity of a compound eliminated by a combination including E_i will be at least as much as the quantity eliminated by E_i alone. A good candidate enzyme at this step is the one that ensures that lesser of the target compounds remain after its inhibition. Also, it should ensure that the non target compounds suffer the minimum possible damage. Our cost model satisfies these requirements. Then, we compute the cost of each enzyme as the dot product of R_i and V. That is, Cost$(E_i) = R_i \cdot V^T$. Suppose that the enzyme (E_j) is with the minimum cost, that is, $j = \arg\min_j \{\text{Cost}(E_j)\}$. Based on the minimum cost, we update the ratio $R = R_j$ and select E_j as the next enzyme to inhibit. Thus, this strategy chooses the next best enzyme to inhibit dynamically.

The cost of finding the best enzyme takes $O(mn)$, where m and n denote the number of enzymes and compounds in the metabolic network respectively. This is because a vector product costs $O(n)$, and $O(m)$ such products are carried out.

3.3. *Filtering strategies*

So far, we have described how OPMET traverses the state space. In this section we propose two filtering strategies to eliminate large portions of the search space quickly while still guaranteeing the optimal solution. The following theorem (proof is not given to save the space) establishes a relationship between the impact of enzymes and their damage.

Theorem 3.1. *Let $E = \{E_1, E_2, \cdots, E_r\}$ be a set of enzymes. Let C_j be a compound in the metabolic network. Let $d_i(C_j)$, $1 \leq i \leq r$, denote the impact of E_i on C_j. If the inhibition of all the enzymes in E stops the production of C_j, then $\sum_{i=1}^{r}(1 - d_i(C_j)) \geq$ 1.* ■

Next, we describe our filtering strategies.

Target Filter: A combination of enzymes can not be the solution if their inhibition does not delete all the target compounds. This is the motivation behind our *Target filter*. The target filter eliminates a bulk of the search space when it is proven that there is no combination of enzymes in this space that can stop the production of all the target compounds (i.e., there is no useful drug target). This filtering strategy is based on Theorem 3.1. Formally, let node $N = ([e_{\pi_1}, e_{\pi_2}, \cdots, e_{\pi_m}], k, d,$ False) be a node in the search space. Let T denote the set of target compounds. Backtrack if

$$\sum_{i=1}^{k}(1 - d_i(C))e_{\pi_i} + \sum_{i=k+1}^{n}(1 - d_i(C)) < 1, \exists C \in T.$$

In this inequality, the first term indicates the impact of enzymes, which are currently part of the solution set, on the target compounds. The second term represents the impact of the remaining enzymes on the target compounds.

Non-target Filter: This filter quickly determines if there is any solution in the subtree with a damage $d < D$, the global cut-off threshold (see Subsection 3.1). This filter utilizes Theorem 3.1 similar to the Target Filter. The idea is as follows. At a given node N, for each target compound, C, we find the minimum number of enzymes, m such that $\sum_{i=1}^{m}(1 - d_i(C))e_{\pi_i} \geq 1$. This gives us the minimum number of enzymes needed to delete C. Let m_{max} be the maximum value of m for any target compound (i.e, we will need at least m_{max} enzymes to delete the entire target compound set). Now, we sort the remaining enzymes (enzymes not considered so far) in the ascending order of their damage values. Let d_{max} be the damage of the enzyme at index m_{max}. If d_{max} in addition to the damage incurred so far is greater than D, we prune the sub tree rooted at N.

4. Experimental results

We verify the biological validity of the proposed algorithm by employing it on known existing drugs. We evaluate the performance of the OPMET algorithm using the following three criteria: *1) Number of nodes generated:* It represents the total number of enzyme combinations tested to complete the search. *2) Optimal node rank:* This indicates the number of nodes explored before the method arrives at the optimal solution.

3) Execution time: This indicates the total time taken by the method to finish the search.

We extracted the metabolic network information of *E. Coli* from KEGG [15]. The metabolic network in KEGG is divided into smaller networks according to their specific functions. We chose six of these networks for our experiments, based on the number of enzymes. We devised a labeling scheme for the networks which begins with 'N' and is followed by the number of enzymes in the network. For instance, 'N20' indicates a network with 20 enzymes. For each network, we constructed query sets of sizes one, two and four target compounds, by randomly choosing compounds from that network. Each query set contains 10 queries.

Qualitative analysis of OPMET: We first evaluate how well the proposed cost model reflects the biological process. We do this by querying well studies drugs in the literature using OPMET. KEGG contains a database of known drug molecules along with the enzymes they inhibit and their therapeutic category. We use the drugs at this database as our benchmarks. Due to space limitation, we report only four of them. The value in parenthesis that starts with letter "D", "C", or "E" (e.g., D02562) is the unique identifier assigned to the corresponding drug, compound, or enzyme respectively in KEGG.

1. Benoxaprofen (D03080). This drug inhibits arachidonate 5-lipoxygenase (E1.13.11.34) which appears in several networks including arachidonic acid metabolism network (hsa00590). In Pharmacology, 5-lipoxygenase inhibitors will decrease the biosynthesis of LTB4 (C02165), cysteinylcontaining leukotrienes LTC4 (C02166), LTD4 (C05951) and LTE4 (C05952). According to our graph model, the removal of 5-lipoxygenase eliminates three of these compounds LTB4, LTC4 and LTD4 in arachidonic acid metabolism network. Inhibition of this enzyme also eliminates five more compounds, namely 5(S)-HPETE (C05356), 5-HETE (C04805), LTA4 (C00909), 20-OH-LTB4 (C04853) and 9(S)-HPOD (C14827). These compounds can be considered as damage in our model. Running OPMET with LTB4, LTC4, LTD4 and LTE4 as the target compound finds LTA4H (E3.3.2.6) and LTC4 synthase (E4.4.1.20) as the optimal enzyme set. The inhibition of these enzymes eliminates only one non-target compound, 20-OH-LTB4 (C04853). OPMET potentially finds a better solution in this experiment than the existing drug as the same compound is eliminated by the existing drug in addition to four other compounds. Indeed, recent research supports our model since the anti-inflammatory effect of the levels of LTA4H [22] and LTC4 [29] have been observed.

2. Rasagiline (D02562). This is an antiparkinsonian drug. It inhibits amine oxidase (E.1.4.3.4) which appears in several metabolic networks. In the histidine metabolism network (hsa00340), the removal of amine oxidase eliminates the compounds Methylimidazole acetaldehyde (C05827) Methylimidazoleacetic acid (C05828) according to our graph model. Levels of pros-methylimidazoleacetic acid has correlation with severity of Parkinson's disease in patients [3,21]. This demonstrates that, our model can predict the intended target well. When OPMET is run on the same network with methylimidazoleacetic acid and the methylimidazole acetaldehyde as the target compounds it finds amine oxidase as the optimal target. This implies that Rasgiline is

Table 1. Average number of nodes generated and Optimal Node Rank of exhaustive search and OPMET with random and dynamic enzyme ordering. Optimal Node Rank is given in parentheses.

Id	Exhaustive Search	Random OPMET	Dynamic OPMET
N14	16,384	4,190 (1,220)	3,273 (3.77)
N17	131,072	58,379 (22,303)	38,973 (2.54)
N20	1,048,576	147,605 (100,845)	78,257 (11.36)

Table 2. Average number of nodes generated and Optimal Node Rank of OPMET with no-filtering (A), non-target filter (B), target filter (C), and both filters (D). Optimal Node Rank is given in parentheses. (E) shows the average execution time (millisecond) for the both filters method.

	N17	N20	N24	N28	N32
A	38973 (2.54)	78257 (11.36)	509278 (55893.25)	158989 (10834.55)	4151032 (3.61)
B	35806 (2.54)	76125 (11.36)	462980 (55103.58)	156956 (10803.00)	1512615 (3.61)
C	415 (2.54)	3496 (11.36)	55987 (56.75)	12735 (10801.90)	1049377 (3.61)
D	394 (2.54)	3428 (11.36)	55865 (6.71)	12710 (10801.90)	1044263 (3.61)
E	529.64	2619.34	46273.13	10025.50	816913.55

targeting the optimal enzyme according to our model.

For Ozagrel (D01683) and Erythromycin acistrate (D02523), running OPMET can find the same target enzyme as the actual drug. (details omitted)

Evaluation of prioritization strategies: We compare our OPMET algorithm with a random ordering of enzymes and an exhaustive search. We do not include our filtering strategies here as the goal is to focus on the enzyme ordering. We present the results only up to a network of size 20 enzymes. This is because, beyond this, the search space grows rapidly, necessitating the use of filtering strategies.

Table 1 shows the average number of nodes generated and the average optimal node rank of OPMET to that of an exhaustive search. The results show that OPMET with dynamic enzyme ordering is the best strategy for all the tested networks. It generates the least number of nodes in all the experiments. All the methods generate significantly large number of nodes for N17. This is because the number of reactions and compounds of this network is much larger than the other networks, resulting in more interactions in the network. OPMET has small Optimal Node Ranks. On an average, it arrives at the optimal solution within the generation of 0.008 % of the number of nodes possible in an exhaustive search. This is significantly better than the random ordering which arrives at the optimal solution within the generation of 11 %.

Evaluation of filtering strategies: We measure how much our filtering strategies reduce the search space. The experiments are performed using OPMET with dynamic enzyme ordering. Table 2 shows the average number of nodes generated, the average optimal node rank and the average execution time for the combined filters. The combined filters show the best pruning. On an average, the combined filters prune 91.5 % of the nodes generated in the method without filters. We also see that most of this benefit is obtained from the Target Filter (it filters 91.4 % of the nodes generated by the method without filters). The combined filter generates only 12700 nodes for N28 (0.004 % of an exhaustive search). All the methods have the same optimal node rank for networks except N24. This suggests that OPMET yielded the optimal solution as early as possible for these networks. For N24, the combined filter shows that filtering strategies can also lead to advancement in finding the optimal solution. For N24, Target filter arrives at the optimal solution 99 % earlier and the combined filters arrive at the optimal solution 99.9 % earlier than the method without filters (the additional 0.9 % improvement is obtained from the non-target filter). We observe that the target filter is more efficient than the non-target filter and the combined filter has the best performance. We observe that there is no clear correlation between the size of the target compound set and the number of nodes explored (results are not shown due to space limitation).

5. Related Work

Classical drug discovery approaches involve incorporating a large number of hypothetical targets into in-vitro or cell-based assays and performing automated high throughput screening (HTS) of vast chemical compound libraries [30,9]. Post-genomic advances in bioinformatics have fostered the development of rational drug-design methods and reduction of serious side-effects [8,5,4]. This has engendered the concept of *reverse pharmacology* [27], in which, the first step is the identification of protein targets, that may be critical intervention points in a disease process [23,1]. The reverse approach is driven by the mechanics of the disease and hence is expected to be more efficient than the classical approach [27].

Rapid identification of enzyme (or protein) targets needs a thorough understanding of the underlying metabolic network of the organism affected by a disease. The availability of fully sequenced genomes has enabled researchers to integrate the available genomic information to reconstruct and study metabolic networks [28,13]. These studies have revealed important properties of these networks [10,2,18]. The potential of an enzyme to be an effective drug target is considered to be related to its essentiality in the corresponding metabolic network [14]. Lemke et. al proposed the measure *enzyme damage* as an indicator of enzyme essentiality [17,19]. Recently, a computational approach for prioritizing potential drug targets for antimalarial drugs has been developed [31]. A choke-point analysis of *P.falciparcum* was performed to identify essential enzymes which are potential drug targets. The possibility of using enzyme inhibitors as antiparasitic drugs is being investigated through stoichiometric analysis of the metabolic networks of parasites [6,7]. These studies show the effectiveness of computational techniques in reverse pharmacological approaches.

A combination of microarray time-course data and gene-knockout data was used to study the effects of a chemical compound on a gene network [12]. An investigation of metabolite essentiality is carried out with the help of stoichiometric analysis [11]. These approaches underline the importance of studying the role of compounds (metabolites) during the pursuit of computational solutions to pharmacological problems.

6. Conclusions

In this paper, we formulated the optimal enzyme-combination identification problem as an optimization problem on metabolic networks. We proposed OPMET, a branch-and-bound algorithm to explore the search space dynamically. We also developed two filtering strategies to prune the search space while still guaranteeing an optimal solution. The filters compute an upper bound to the number of target compounds deleted and a lower bound to the side-effect respectively.

Our experiments on the human metabolic network demonstrates that the proposed model can accurately identify the target enzymes for known successful drugs in the literature. More specifically, OPMET found the same target enzyme as Rasagiline, Ozagrel, and Erythromycin acistrate when their target compounds are given as input. OPMET found a different set of enzymes than Benoxaprofen for the target compounds of Benoxaprofen. OPMET's solution in this case has a great potential to be better than Benoxaprofen since OPMET's solution damages only one non-target compound whereas Benoxaprofen damages five non-target compounds including the compound damaged by OPMET's solution. Our experiments also show that OPMET can reduce the total search time by several orders of magnitude as compared to the exhaustive search. The optimal solution is reached by OPMET within the exploration of 0.005 % of the total search space on an average, proving that our methods are effective in approximating the impact of an enzyme on a compound. OPMET with combined filters pruned 91.6 % of the search space on average.

References

1. 'Proteome Mining' can zero in on Drug Targets. Duke University medical news, Aug 2004.
2. Masanori Arita. The metabolic world of Escherichia coli is not small. *PNAS*, 101(6):1543–1547, Feb 2004.
3. P. Blandina and G. Cherici et al. Release of glutamate from striatum of freely moving rats by pros-methylimidazoleacetic acid. *Journal of Neurochemistry*, 64(2):788–793, 1995.
4. Samuel Broder and J. Craig Venter. Sequencing the Entire Genomes of Free-Living Organisms: The Foundation of Pharmacology in the New Millennium. *Annual Review of Pharmacology and Toxicology*, 40:97–132, Apr 2000.
5. Sumit K. Chanda and Jeremy S. Caldwell. Fulfilling the promise: drug discovery in the post-genomic era. *Drug Discovery Today*, 8(4):168–174, Feb 2003.
6. A. Cornish-Bowden. Why is uncompetitive inhibition so rare? : A possible explanation, with implications for the design of drugs and pesticides. *FEBS Letters*, 203(1):3–6, Jul 1986.
7. A. Cornish-Bowden and J. S. Hofmeyr. The Role of Stoichiometric Analysis in Studies of Metabolism: An Example. *Journal of Theoretical Biology*, 216:179–191, May 2002.
8. Eugene J. Davidov, Joanne M. Holland, Edward W. Marple, and Stephen Naylor. Advancing drug discovery through systems biology. *Drug Discovery Today*, 8(4):175–183, Feb 2003.
9. J Drews. Drug Discovery: A Historical Perspective. *Science*, 287(5460):1960–1964, Mar 2000.
10. Vassily Hatzimanikatis, Chunhui Li, Justin A Ionita, and Linda J Broadbelt. Metabolic networks: enzyme function and metabolite structure. *Current Opinion in Structural Biology*, 14(3):300–306, 2004.
11. Marcin Imielinski, Calin Belta, Adam Halasz, and Harvey Rubin. Investigating metabolite

essentiality through genome scale analysis of E. coli production capabilities. *Bioinformatics*, Jan 2005.

12. S. Imoto and Y. Tamada et. al. Computational Strategy for Discovering Druggable Gene Networks from Genome-Wide RNA Expression Profiles. In *PSB*, 2006.
13. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabasi. The large-scale organization of metabolic networks. *letters to NATURE*, 407:651–654, Oct 2000.
14. Hawoong Jeong, Zoltan N. Oltvai, and Albert-Laszlo Barabasi. Prediction of Protein Essentiality Based on Genomic Data. *ComPlexUs*, 1:19–28, 2003.
15. M Kanehisa and S Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28(1):27–30, Jan 2000.
16. S.A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
17. Ney Lemke, Fabiana Herdia, Cludia K. Barcellos, Adriana N. dos Reis, and Jos C. M. Mombach. Essentiality and damage in metabolic networks. *Bioinformatics*, 20(1):115–119, Jan 2004.
18. Hong-Wu Ma, Xue-Ming Zhao, Ying-Jin Yuan, and An-Ping Zeng. Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph. *Bioinformatics*, 20(12):1870–1876, 2004.
19. J. C. Mombach and N. Lemke et al. Bioinformatics analysis of mycoplasma metabolism: Important enzymes, metabolic similarities, and redundancy. *Computers in Biology and Medicine*, 2005.
20. M.T.A. Ocampo and W. Chaung et. al. Targeted deletion of mNth1 reveals a novel DNA repair enzyme activity. *Mol Cell Biol.*, 22(17):6111–21, Sep 2002.
21. G. D. Prell, J. K. Khandelwal, R. S. Burns, P. Blandina, A. M. Morrishow, and J. P. Green. Levels of pros-methylimidazoleacetic acid: Correlation with severity of Parkinson's disease in CSF of patients and with the depletion of striatal dopamine and its metabolites in MPTP-treated mice. *Journal of Neural Transmission*, 3(2):1435–1463, 1991.
22. Navin L. Rao and Paul J. et al. Dunford. Anti-Inflammatory Activity of a Potent, Selective Leukotriene A4 Hydrolase Inhibitor in Comparison with the 5-Lipoxygenase Inhibitor Zileuton. *J Pharmacol Exp Ther*, 321(3):1154–1160, 2007.
23. C Smith. Hitting the target. *Nature*, 422:341–347, Mar 2003.
24. R. Somogyi and C.A. Sniegoski. Modeling the complexity of genetic networks: Understanding multi-gene and pleiotropic regulation. *Complexity*, 1:45–63, 1996.
25. Padmavati Sridhar, Tamer Kahveci, and Sanjay Ranka. An iterative algorithm for metabolic network-based drug target identification. In *PSB*, volume 12, pages 88–99, 2007.
26. R. Surtees and N. Blau. The neurochemistry of phenylketonuria. *European Journal of Pediatrics*, 159:109–13, 2000.
27. T. Takenaka. Classical vs reverse pharmacology in drug discovery. *BJU International*, 88(2):7–10, Sep 2001.
28. S. A. Teichmann and S. C. G. Rison et al. The Evolution and Structural Anatomy of the Small Molecule Metabolic Pathways in Escherichia coli. *JMB*, 311:693–708, 2001.
29. M. J. Torres-Galvan and N. Ortega et al. LTC4-synthase A-444C polymorphism: lack of association with NSAID-induced isolated periorbital angioedema in a Spanish population. *Ann Allergy Asthma Immunol.*, 87(6):506–10, 2001.
30. Gunther Wess. How to escape the bottleneck of medicinal chemistry. *Drug Discovery Today*, 7(10):533–535, May 2002.
31. I. Yeh and T. Hanekamp et al. Computational Analysis of Plasmodium falciparum Metabolism: Organizing Genomic Information to Facilitate Drug Discovery. *Genome Research*, 14:917–924, 2004.