Protein protein interaction network evaluation for identifying potential drug targets

Fereydoun Hormozdiari *†, Raheleh Salari *†, Vineet Bafna ‡, S. Cenk Sahinalp †

December 15, 2009

Abstract

As pathogens evolve effective schemes to overcome the effect of antibiotics, the prevalent “one drug and one drug target” approach is falling behind. We propose novel strategies for identifying potential multiple-drug targets in pathogenic PPI networks with the goal of disrupting known pathways/complexes.

Given a set $S$ of pathogenic pathways/complexes, we first consider computing the minimum number of proteins (with no human orthologs) whose removal from the PPI network disrupts all pathways/complexes. Unfortunately even the best approximation algorithms for this (NP-hard) problem return too many targets to be practical. Thus we focus on computing the optimal tradeoff (i.e. maximum ratio) between the number of disrupted essential pathways/complexes and the protein targets. For this “sparsest cut” problem, we describe two polynomial time algorithms with respective approximation factors of $|S|$ and $O(\sqrt{n})$ ($n$: number of nodes). On the E.coli PPI network with 9 essential (signaling) paths from the KEGG database, our algorithms show how to disrupt 3 of them by targeting only 3 proteins (2 of them essential proteins).

We also consider the case where there are no available essential pathways/complexes to guide us. In order to maximize the number of disrupted “potential” pathways/complexes we show how to compute the smallest set of proteins whose removal partitions the PPI network into two almost-equal sized subnetworks so as to maximize the number of potential pathways/complexes disrupted. This approach yields 28 potential targets (4 of them known drug targets) on the E.coli PPI network whose removal partitions it to two subnetworks with relative sizes of 1 to 5.

1 Introduction

With an increase in drug-resistant pathogenic strains, infectious diseases are on the rise. Infectious diseases are already a leading cause of productivity loss and are responsible for roughly a third of annual deaths worldwide; sepsis and mortality caused by infectious diseases are also on the rise in developed countries like the U.S. Additionally, newly emerging diseases like the avian flu are causing considerable concern: a new global pandemic could have a significant social and economic impact [28].

Bacteria evolve effective schemes to overcome the effect of antibiotics. For example, many common antibiotics, such as tetracycline, target the ribosome. Bacteria develop resistance to such

---

*These authors contributed equally to the paper.
†School of Computing Science, Simon Fraser University, Canada.
‡Department of Computer Science and Engineering, University of California San Diego, USA.
antibiotics by up regulating molecular pumps that force the antibiotic from the cell, as well as proteins that shield the ribosomes. As bacteria continue to evolve resistance mechanisms, research on developing antibiotics with a novel, single mechanism of action has become a continuing process with no end in sight.

The prevalent approach to drug development is based on a principle established almost 100 years ago by Paul Ehrlich’s pioneering search for magic bullets that could selectively target the constituents of infectious organisms relative to the hosts. Ehrlich’s work resulted in a gradual shift from the use of complex extracts to the use of defined small molecules and toward today’s single target therapies.

The current “rational drug design” approach aims to target a single essential molecule of the pathogen. This is often a protein (with no human orthologs [15]) with an essential role in virulence or in the pathogen’s survival; examples include receptors, enzymes and ion channels [38]. Unfortunately, only a small percent (<10%) of pathogenic genes are known to be essential; this ensures the organism’s resilience against genetic modification. Furthermore many putative targets might have “backup” systems, hence the desired structural/functional effect may not emerge from solely attacking a single target [36, 6].

An emerging approach in drug target identification is the result of newly developed multi-target drugs, in contrast to the prevalent single-target drugs. More and more, novel therapies for complex diseases such as AIDS, atherosclerosis and cancer are becoming “combinatorial”, i.e. they employ multi-target drugs [6, 18, 27, 3]. Thus, many researchers now argue that attacking multiple targets provide a promising direction in drug development against infectious diseases [9, 30, 22, 36, 29, 42].

An inspiration for targeting multiple proteins in drug design comes from synthetic lethality [36]: here, two mutations which do not imply drastic phenotypes independently can have a lethal effect only if they occur together. Synthetic lethality might be result of two proteins being non-essential components of the same complex, or being a part of two separate pathways that meet at an essential end point of a function. Other naturally occurring multi-target systems include snake and spider venom which have multiple targets to ensure lethality. On the defensive side, some plants have a combative-strategy to defend themselves against pathogenic attacks [6].

In this paper, we explore computational strategies for identifying multiple potential drug targets in a pathogen. We aim to identify targets on the protein protein interaction (PPI) networks of pathogens so that several “essential pathways/complexes” are disrupted as much as possible; i.e. there could be no backup paths on the PPI network.

A PPI network involves nodes which correspond to proteins and edges which denote interactions among proteins. High-throughput technologies now allow rapid identification of protein protein interactions and networks they imply. Databases containing networks of some pathogenic microbes are now available, and others are being developed rapidly [34, 40].

One approach to drug target identification on essential networks aims to identify their “weak” points: PPI networks are generally accepted to be scale free and such networks typically have a few “hubs” (nodes with high degree). The hubs play a central role in the network, connecting several nodes to others; thus their removal, in theory, may disrupt a number of essential pathways/complexes. As a result hubs may be considered for further evaluation for being potential drug targets [17]. Unfortunately, it turns out that the degree of a node is only weakly correlated to the lethality of removing the associated protein from the PPI network and thus its potential as a drug target [7]. Another strategy for discovering weak points in a network may consider “more global” properties of a node such as its betweenness or closeness [10, 7]. Nodes with high
betweenness and/or closeness could be picked as initial candidates for drug targets. One can also employ flux-balance analysis of metabolic networks to find weak points of a pathological metabolism towards discovering effective drug targets [20].

All the above, conceptually simple approaches aim to identify proteins which are “weak points of essential networks” as suggested by their topological properties in the PPI network. However, none of them directly aim to solve the “central problem” in drug target identification, i.e. disabling several essential functions by disrupting specific pathways/complexes (including all possible backup paths) through attacking the fewest number of proteins. In fact, we show that some of these approaches could suggest targets that are far from solving this problem. In this paper we will focus on direct strategies that aim to solve the central problem in drug target prediction as described below.

1.1 Our approach and a summary of our results

The main goal of this paper is to find a small number of protein targets in a pathogen, whose deactivation/silencing can disable or kill the infectious agent, without making any noticeable damage to the host. The “removal” of a set of the proteins on an essential pathway/complex, would disrupt the pathway/complex and prohibit the pathogen from performing a vital function, disturb its regular cell cycle and possibly kill it [37, 32]. Note that to reduce the possibility of a pathogen developing resistance against a drug, one must aim to simultaneously target as many pathways/complexes as possible.

The key problem associated with this approach is that some of the proteins in a pathway/complex may have backups: there may be alternative routes in the pathway/complex that bypass the protein in consideration. Consider, for example, the bacterial chemotaxis pathway in H.pylori (Figure 1). Looking at the pathway alone one may consider to target any one of the proteins in the pathway, for instance CheY. However, in the PPI network of H.pylori (Figure 1) one can notice three independent paths between MCP (tlpA gene) and MotB. Notice that these paths go through proteins CheW, smpB and HP0249 respectively; targeting all these proteins prohibits the possibility of any (backup) paths between MCP and MotB.

Note that in order to prohibit any side-effect on the host organism, the set of proteins identified as potential drug targets should have no orthologs in the host. This introduces additional constraints on the problem. The other type of constraints is related to the targetability of proteins. The set of identified target proteins should satisfy the targetability requirements.

In this paper we propose two computational methods to identify multiple potential drug target candidates in a pathogen that avoid the above mentioned problems associated with conventional target identification strategies. The first method aims to disrupt all-maximum number of available essential pathways/complexes by removing the minimum possible number of nodes from the PPI network. Note that available PPI networks, especially of well known pathogens, only provide an approximation to the complete PPI network, as they are typically obtained through co-expression analysis (yielding many false positives and negatives). Even more reliable PPI networks whose interactions are verified via two hybrid assays in a semi in vivo environment may have not only false positives but also may miss existing interactions, due to the high cost of verifying each possible interaction. Thus the above strategy needs to disrupt as many available pathways as possible for increasing its chance to be lethal.

Our first method is useful for pathogens for which number of essential pathways/complexes are known and the PPI network is reasonably well developed. Unfortunately in other organisms very few
(or no) essential pathways/complexes are available. In such a case, one can aim to target a number
of proteins whose removal from the network disrupts as many “potential pathways” as possible.
In the lack of available pathways/complexes, every pair of nodes provide a pair of end nodes of
a “potential pathway/complex”. Thus, to maximize the number of potential pathways/complexes
disrupted, our second method aims to compute the minimum number of nodes whose removal from
the network partitions it into two disconnected components of “roughly equal” size.

Finally further pruning based on the specific biological information can be applied to the set
of potential targets. This biological knowledge can come from structural features of proteins, the
essentiality of one or a group of proteins, and additional information about the given pathways and
complexes. For example, if one of the returned potential targets is known that it can not be part
of a (backup) path in given essential pathways or complexes, it should be removed from the target
list.

Target identification with the aid of essential pathways and complexes. Given a set of
essential pathways/complexes, our first approach aims to disrupt as many communication paths as
possible between each possible pair of end nodes of essential pathways/complexes with the minimum
number of proteins that need to be targeted. A pair of proteins in a pathway/complex are considered
to be end nodes, if the number of hops between them is equal to the diameter (i.e. the maximum
distance between two nodes) of the pathway/complex. Disrupting all the paths between such a
pair in the PPI network will make sure that they will have no means of “communicational” and
thus the function associated with the pathway/complex will be disrupted as well.

The problem can be modeled as the well known multicut problem which is NP-hard but there
are a number of approximation algorithms towards its solution. Unfortunately the size of the cut
set obtained even by the best approximation algorithms is typically too large to be practical. As a
result we focus on obtaining the “best” possible tradeoff between the number of disrupted essential
pathways/complexes and the proteins that need to be targeted with the goal of maximizing the
ratio between these two quantities. This problem is known as sparsest cut problem. Although the
sparsest cut problem is NP-hard as well, we describe two approximation algorithms for solving it
(see section Methods). The first algorithm (DSC ) is conceptually very simple, fast and effective
in practice, but it works only when there are a few pathways/complexes; the second one is not as
fast but its running time is independent of the number of pathways/complexes. On the E.coli PPI
network with 9 essential (signaling) paths from the KEGG database [19], our algorithms identified
3 proteins (2 of them essential proteins) as potential targets, whose removal can disrupt 3 essential
pathways/complexes (including all possible backup paths).

Target identification without the aid of essential pathways and complexes. In case we
have limited or no information on the essential pathways/complexes of a pathogen, a new strategy
need to be developed. For this case we aim to compute a set of drug targets which will disrupt the
maximum number of “potential” pathways/complexes. As every pair of nodes can be potential end
nodes of an unknown pathway/complex our goal is to maximize the number of pairs of nodes which
can not communicate when nodes of a “cut” are all removed. Thus our goal is to compute the
smallest set of nodes whose removal from the network partitions it to two disconnected subnetworks
with roughly equal size (the number of node pairs separated is maximized when the partition is
balanced).

Our general approach can be formulated as a minimum weighted node separator problem. Al-
though computing a minimum weighted node separator is a known NP-hard problem, there are
a number of approximation algorithms available in the literature. Unfortunately, many of these
approximation algorithms perform poorly in practice, especially when dealing with sizable graphs
such as pathogenic PPI networks [24]. In order to “solve” the minimum weighted node separator
problem we thus introduce a number of heuristic approaches, mainly a heuristic called HMWS (see
subsection Minimum Weighted Separator).

We compared our approach with alternative simple heuristics, e.g. one which iteratively removes
hubs or that which iteratively removes nodes with the highest betweenness value. Our method
always performed better than these alternatives for all values of the balance factor $\beta$, i.e. the
number of targets it returns is always fewer than that obtained by the alternatives. In fact, our
general approach returned 28 potential targets on the E.coli PPI network whose removal partitions
it to two subnetworks with relative sizes of 1 to 5. Among the 28 potential targets discovered, 4 are
known drug targets (the total number of known E.coli drug targets in approved drugs in drugbank
is 15).

2 Methods

Unfortunately the multicut, the sparsest node cut and the minimum weighted node separator
problems which are employed in our two strategies are all NP-complete [26, 11]. For solving the
first two problems we consider approximation algorithms. We first describe a polynomial time
$O(|S|)$ factor approximation algorithm for the multicut problem where $S$ is the set of all end node
pairs on available pathways/complexes. This solution to the multicut problem is then used to obtain
a simple combinatorial algorithm for the sparsest node cut problem again with an approximation
factor of $O(\sqrt{n})$ with a running time of $O(2^{|S|} \cdot \text{poly}(n))$. This running time is polynomial in $n$ if $|S| = O(\log n)$. We also describe a second solution to the sparsest node cut problem based on
linear programming which has an approximation factor of $O(\sqrt{n})$ and runs in time polynomial
with $n$. These algorithms are all given in subsection Multicut and Sparsest Cut. For the minimum
weighted node separator problem, even the best available approximation algorithms may turn out
to be intractable for sizable graphs; thus we describe a number of heuristic approaches in subsection
Minimum Weighted Node Separator.

2.1 Target identification with the aid of essential pathways and complexes

Given a set of essential pathogenic pathways/complexes, we aim to compute the minimum number
of proteins that need to be targeted so as to disrupt as many essential pathways/complexes as
possible. We model our problem with the well known weighted node sparsest cut problem as
follows. Let $G(V, E)$ be an undirected graph where $w_v$ denotes the weight of a node $v$; we will set
$w_v = 1$ if the protein represented by node $v$ does not have an ortholog in the host organism and
$w_v = \infty$ otherwise (A more sophisticated weighting schema can also be utilized, where proteins
such as surface proteins, which can be targeted more easily, get lower weights.).

Now let $S = \{(s_i, t_i) : s_i, t_i \in V\}$ be a set of node pairs in $V$ where each pair represents a pair
of proteins that are a part of at least one common essential pathway/complex. We are also given
a real valued essentiality function defined on each pair of proteins in $S$ ($ess : S \rightarrow \mathbb{R}_{>0}$) which
represents how essential a pathway/complex is for pathogen’s survival.

The node weighted sparsest cut problem in a network aims to find a cut $C$ which minimizes the
ratio $\frac{W(C)}{ess(C)}$, where $W(C)$ is total weight of nodes in the cut, and $ess(C)$ is total essentiality of all $(s_i, t_i)$ pairs which are separated by the cut.

In this section we describe two approximation algorithms for solving the node sparsest cut problem. Our first method (which we call DSC) is a very simple combinatorial approach that solves a multicut problem for each subset of pathways/complexes and picks up the solution that gives the best ratio. We show that this simple approach has an approximation factor of $O(\log n)$; and its running time is polynomial only when the number of pathways/complexes is limited to $O(\log n)$. The second method has a relatively large approximation factor of $O(\sqrt{n})$ however it does not have any limitation on the number of pathways/complexes (because of its linear programming approach it is referred to as LP). The first algorithm requires a solution to the multicut problem, which can be employed for identifying potential drug targets as well. Both solutions require transforming the input given as an undirected node-weighted network to a directed edge-weighted network (as per [24]) as follows.

Transforming an undirected node-weighted network to a directed edge-weighted network Given an undirected graph $G(V, E)$ with node weight $\pi : V \rightarrow \mathbb{R}_{>0}$ as input, we generate a directed graph $G' = (V', E')$ with edge weights $w : E' \rightarrow \mathbb{R}_{>0}$ as follows. $V'$ contains each node $v$, as well as a copy $v'$. The edge-set $E'$ contains a directed edge from each node $v$ to its copy $v'$. Additionally, for all $(u, v) \in E$, we add $(u', v), (v', u)$ to $E'$. Thus, $E' = \{(v, v')|v \in V\} \cup \{(u', v)|(u, v) \in E\} \cup \{(v', u)|(u, v) \in E\}$. Weights are assigned to each edge so that $w(v, v') = \pi(v)$ and $w(u', v) = \infty$ for $v \neq u$.

The sparsest cut problem on graph $G'$ can now be defined as follows.

The Sparsest Cut Problem

Input: A directed graph $G'(V', E')$, with edge weights $w : E' \rightarrow \mathbb{R}_{>0}$, and a set $S$ of $k$ end node pairs $\{(s_i, t_i)|s_i, t_i \in V'\}$ where a positive “essentiality” value $e : S \rightarrow \mathbb{R}_{>0}$ is associated with each pair.

Problem: Compute

$$C = \arg\min_{C \subseteq V'} \frac{W(C)}{ess(C)} = \arg\min_{C \subseteq V'} \frac{\sum_{u,v \in C} w(u,v)}{\sum_{(s_i, t_i) \in S} ess(i)}$$

Thus the sparsest cut on $G'$ is a set of edges whose removal maximizes the ratio of the total essentiality of the end node pairs disconnected and the total weight of edges removed. Note that an edge cut in $G'$ corresponds to a node cut in $G$, whose removal maximizes the total essentiality of the end node pairs disconnected in $G$ and the total weight of the nodes removed. This provides the best possible tradeoff between the number of proteins to be targeted and the number of essential pathways/complexes that would be disrupted.

We describe two algorithms for solving the edge sparsest cut problem. The first one is a simple combinatorial algorithm which has a running time of $O(n^3 \log^2 n)$, provided $|S| = O(\log n)$. The approximation factor for this algorithm is $O(\log n)$. For the case that the number of pathways/complexes (or the number of end node pairs) is large, we give another algorithm based on Linear Programming which has a (worse) approximation factor of $O(\sqrt{n})$[14].
2.1.1 Solving the Multicut and Directed Sparsest Cut Problems via a Combinatorial Approach

The minimum multicut problem asks to find the minimum number of nodes that need to be removed from a network so as to disconnect all end node pairs of a given set of pathways/complexes. Below we describe the MLC procedure, which gives an approximation to the minimum multicut problem within a factor of $|S|$ where $S$ is the set of all end node pairs in each pathway/complex we consider. As the number of nodes that can be returned by the procedure MLC can be too large for practical purposes, we consider every possible subset $\tilde{S}$ of our collection of pathways/complexes $S$ and obtain via MLC the cut set which gives the maximum ratio between the number of end node pairs disconnected and the size of the cut set. It turns out that this cut set provides an $|S|$-approximation to the directed sparsest cut problem. Provided that the number of end node pairs is $O(\log n)$, the running time of this approach to solve the directed sparsest cut problem will be polynomial with $n$ and would give an approximation factor of $O(\log n)$.

Procedure MLC($\tilde{S} \subseteq S$)

1. For each $(s_i, t_i) \in \tilde{S}$, compute $E_i = \text{Min-s-t-Cut}(s_i, t_i)$.
2. Return $\cup_i E_i$.

As each $E_i$ is a lower-bound on the optimal cut, MLC provides a factor $|S|$ approximation to the minimum multicut problem. The implementation of min-s-t-cut [13] runs in $O(n^2 \log n)$ time. In each run of the MLC procedure, we have at most $|S|$ pairs, so the running time of finding minimum multicut is $O(|S|n^2 \log n)$. Restricting $|S| \leq \log n$, we obtain a log $n$ approximation guarantee with running time $O(n^3 \log^2 n)$.

We implemented the procedure MLC which utilizes a procedure for the min-s-t cut proposed by Goldberg and Tarjan [13] as well as the directed sparsest cut algorithm in C++. On the E.coli PPI network with 1440 proteins and 5871 interactions, and 9 end node pairs from 4 essential signaling pathways and 1 complex, our software was able to compute potential targets in less than 5 minutes on a high end PC.

2.1.2 Solving the Sparsest Cut Problem via Linear Programming

One way to solve directed sparsest cut problem is by means of linear programming (LP). The following LP algorithm gives an $O(\sqrt{n})$ approximation [14]. Given a graph $G$ it assigns a length $d$ to each edge in a way that the total length assigned to edges is small, however the distance between end node pairs is large. Here $dst(u, v)$ represents the shortest distance from $u$ to $v$.

minimize:
$$ W(d) = \sum_{e \in E'} w(e)d(e) $$
subject to:
$$ \sum_i \text{dst}(s_i, t_i) \cdot \text{ess}(i) = 1 $$
$$ \text{dst}(u, v) + d(v, x) \geq \text{dst}(u, x) \quad u, v, x \in V', (u, x) \in E' $$
$$ d(u, v) \geq 0 \quad (u, v) \in E' $$
$$ \text{dst}(u, v) \geq 0 \quad u, v \in V' $$

The approximation algorithm for the sparsest cut problem assigns a length value $l$ to each edge based on the $d$ value from the above LP (where all edges in the cut have the largest $l$ value among all edges.
in $G$). Define the max-essentiality-separation as $S_{\text{max}}(l) = \sum_{i} \text{ess}(i) \min_{p \in P_{s_{i}, t_{i}}} \{\max_{e \in p} d(e)\}$, where $P_{s_{i}, t_{i}}$ is the set of all the paths from $s_{i}$ to $t_{i}$. The following polynomial time rounding algorithm finds the cut $C$ whose sparsity is at most $W(l)/S_{\text{max}}(l)$ which is less than $O(\sqrt{n})W(d)/S_{\text{max}}(d)$. Since LP is a relaxation to the sparsest cut problem, cut $C$ has sparsity within a factor of $O(\sqrt{n})$ of the optimal solution.

### Rounding Algorithm

1. For all edges $e$, initiate the length $l(e) = 0$.
2. For each $(s_{i}, t_{i}) \in S$: Identify every edge $e$ that lie on some $s_{i} - t_{i}$ path and have $d(e) \geq \frac{\text{dist}(s_{i}, t_{i})}{\sqrt{n}}$, and set $l(e) = \text{dist}(s_{i}, t_{i})$.
3. Group all the edges into $E_{1}, ..., E_{k}$ according to their assigned length $l$ in decreasing order.
4. Compute $\Delta W = \sum_{e \in E_{1}} (l_{E_{1}} - l_{E_{2}}) w(e)$ and $\Delta S = \sum_{s_{i}, t_{i}}$ separated by $E_{1} (l_{E_{1}} - l_{E_{2}}) \text{ess}(i)$. If $\Delta W(l)/\Delta S(l) > W(l)/S_{\text{max}}(l)$, we scale all edges in $E_{1}$ down to length $l_{E_{2}}$ and repeat this step, otherwise return $E_{1}$ as $C$.

We implemented the above rounding algorithm in C++ and executed the above linear program via the well known COIN-OR LP tool [25]. This method is much slower than the previous one: on the E.coli PPI network with 9 end node pairs, it took about 10 hours to compute the targets on a SUN Fire X4600 server with 8 Dual AMD Opteron processors of 2.6GHz and 64GB of RAM.

#### 2.2 Target identification without the aid of essential pathways and complexes

In case we have limited or no information on the essential pathways/complexes, our goal is to compute the smallest set of nodes whose removal from the network partitions it to two disconnected subnetworks with roughly equal size (the number of node pairs separated is maximized when the partition is balanced). This approach can be formulated as a minimum weighted node separator problem.

Given a graph $G(V, E)$ ($|V| = n$ and $|E| = m$) with each node $v$ having a weight $w_{v}$, let $C$ be a set of nodes whose removal from $G$ partitions it into two disconnected set of nodes $(S, \bar{S})$. $C$ is called a $\beta$-balanced separator if $\min\{|S|, |\bar{S}|\} \geq \beta n$ ($\beta < 0.5$). The weight of cut $C$ is defined to be the total weight of the nodes in $C$; more formally, $W(C) = \sum_{v \in C} w_{v}$. Given a constant $\beta$, the minimum weighted node separator problem, asks to find a $\beta$-balanced separator of the given graph $G(V, E)$ where weight of the cut $W(C)$ is minimized. We can make sure that the target set includes no proteins with human orthologs by setting $w_{v}$, the weight of a node $v$ as 1 if the pathogenic protein represented by $v$ does not have a human ortholog and as $\infty$ otherwise. In fact further decreasing weights of nodes associated with proteins that are easier to target or are known to be more effective drug targets may improve the chance that such proteins are indeed included in the target set.

The first approximation algorithm for the minimum weighted node separator problem was described by Leighton and Rao [24], and is based on reducing this problem to a multicommodity flow problem. The approximation factor achieved by this algorithm is $O(\log n)$. This result was recently improved in [8, 2] which describes an $O(\sqrt{\log n})$ approximation algorithm using semidefinite programming. Because the underlying algorithms of both methods are computationally intensive,
they are intractable for large graphs (e.g. the multicommodity flow algorithm proposed in [23], failed to return a result on the PPI networks we considered after two weeks of execution on a high end 64-bit SUN server with 16 dual core processors sharing a 64GB of RAM).

It is also possible to use heuristic approaches to obtain a node separator, which hopefully will partition the input network into roughly equal size subnetworks. One such natural heuristic, for example, would iteratively choose the highest degree node available and remove it from the network until a cut is obtained. Unfortunately our experiments with this heuristic returned cuts sets which would be too large for practical purposes.

As a result we proposed alternative heuristics which implicitly exploit the fact that known PPI networks have power law degree distributions. Our approach is based on the notion of betweenness, which measures the centrality of a node \( v \) in a graph \( G \). The standard definition of the betweenness of a node \( v \) in graph \( G \) is the fraction of the shortest paths in \( G \) (between all possible node pairs) which go through \( v \). A more general notion, which not only focuses on shortest paths but all possible paths between pairs of nodes is the random walk betweenness [31] which counts how often \( v \) is visited by a “random walk” between each pair of nodes.

Formally speaking, the random walk betweenness of a node \( v \) with end node pairs \( s \) and \( t \) is equal to the probability that \( v \) is visited by a random walk between \( s \) and \( t \). The overall random walk betweenness of a node \( v \), denoted \( RWBet(v) \), is the probability that \( v \) is visited by random walks between all possible pairs of nodes \( s \) and \( t \).

The algorithm we use for calculating random walk betweenness of nodes is based on known techniques from linear algebra [31]; due to space limitations we skip the details here. We just note that the algorithm can calculate the random walk betweenness for all nodes in \( O(n^3) \) time on a sparse graph (known PPI networks are sparse) and it requires space of \( O(n^2) \). In fact different centrality metrics can be used for calculating the initial separator, the reason that we selected random walk betweenness is because of its ability to count essentially all paths between vertices. Moreover, experimentally the greedy algorithm based on random walk betweenness, for a fixed \( \beta \), returns smaller separator in compare to other centrality metrics. Following the computation of \( RWBet(v) \) values for all \( v \), we compute the Minimum Node Separator of \( G \) as follows.

**A Heuristic Approach to the Minimum Weighted Node Separator Problem (HMWS)**

Our approach has three phases: **Split**, **Merge & Cut**. **Split**\((G, \beta)\) returns an initial-cut \( I_C \) such that each connected component obtained after removing \( I_C \) has fewer than \((1 - \beta)n\) nodes. Let \( V \) be the set of connected components of \( G \). Merge\((G, \beta, V)\) partitions the components in \( V \) into two sets \( P_1, P_2 \), such that each set has at least \( \beta n \) nodes. Cut\((G, \beta, S, T)\) re-computes a separator/cut of size at most \(|I_C|\) nodes, such that each partition has at least \( \beta n \) nodes.

**Procedure Split**\((G, \beta)\)

1. Compute \( RWBet(v) \) for all nodes \( v \in V \).
2. Select a large connected component \( V' \) in \( G \) (\(|V'| \geq (1 - \beta)n\)).
3. Select \( v = \arg \max_{v \in V'} RWBet(v) \).
4. \( I_C = I_C + \{v\}; \ V = V' - \{v\} \); Recompute \( V \).
5. If all components \( V_j \) are small, return \( V = \cup_j V_j \); else go to 2.
After running Split, $G$ is partitioned to a set of small components, however our satisfactory goal is to partition the graph to only two balance components. Applying Merge results in two sets of small components that can be reconnected. To perform the Merge step, we use the $\epsilon$-approximation algorithm for subset sum problem [33]. The subset sum problem is a special case of the knapsack problem and is defined as follows: given a set of positive integers \{a_1, ..., a_n\}, and a positive integer $B$, find a subset of the $a_i$ such that their sum is as close as possible to $B$, without exceeding $B$.

**Procedure Merge**($G, \beta, V$)

1. Set $B = (1 - \beta)n$.

2. Return $(S, T)$; where $S$ is the set of nodes associated with result of Subset-Sum ($B, |V_1|, |V_2|, \ldots$) ($\forall V_j \in V$), and $T = V - (S \cup I_C)$.

Now we have two component $S$ and $T$ with separator $I_C$. In step Cut we look for a smaller separator. The new separator is calculated as a subset of not only $I_C$, but also really high degree nodes (top 10%). Here, we give an option to high degree nodes to be selected as target. Note that selecting the nodes in $I_C$ is done in a greedy manner based on centrality properties; however in Procedure Cut by using st-mincut, the overall effect of nodes in the separator is considered.

**Procedure Cut**($G, \beta, S, T$)

1. Add two dummy nodes $s$ and $t$ to the graph.

2. Connect $s$ (respectively, $t$) to the $\beta n$ lowest degree nodes in set $S$ (respectively, $T$).

3. For all nodes $v$ adjacent to $s, t$, weight($v$) = $\infty$.

4. Return node st-mincut (this calculates minimum weighted node separator between $s$ and $t$ [13]) $I$.

As $I_C$ is a candidate separator for $S$ and $T$, we have $|I| \leq |I_C|$. Furthermore, as all nodes connected to $s$ (respectively, $t$) stay in $S$ (respectively, $T$), $|S| \geq \beta n$, and $|T| \geq \beta n$. The running time of above algorithm is $O(n^3)$, and the bottleneck of the algorithm is the random walk betweenness for each node, in the Split phase.

We implemented the procedure for computing the random walk betweenness of all nodes in a PPI network, as well as the Split, Merge and Cut phases of our heuristic method in C++. The calculation of random walk betweenness requires inverting an adjacency matrix, which is done via Mathlab. On the E.coli network, the random walk betweenness of all nodes can be calculated in less than 30 minutes on a high end PC. The implementation of the heuristic takes only seconds to compute potential drug targets.

### 3 Results and Discussion

#### 3.1 Network datasets

The experimental results we report here were performed on the well investigated pathogenic PPI networks of *Escherichia coli* (*E.coli*) and *Helicobacter pylori* (*H.pylori*). *E.coli* is a very well known
infectious agent, and can cause several intestinal and extra-intestinal infections such as urinary tract infections, meningitis, peritonitis, mastitis, septicemia and Gram-negative pneumonia. *H. pylori* causes one of the most common infections. In developed nations up to half of all individuals are infected with this pathogen, and this number rises to more than 90% in many developing nations. Although infected individuals are often asymptomatic, *H. pylori* is a casual agent of chronic gastritis in humans and is strongly associated with the development of both duodenal and gastric ulcers and the rare cancer gastric mucosa-associated lymphoid tissue lymphoma [5].

The above pathogens have the best developed PPI networks among all pathogens; the PPI networks are available through public resources such as the database of interacting proteins (DIP [34, 40]). We used the largest connected components of the PPI networks of both organisms for our experiments. In the rest of the paper, for simplicity we denote the largest connected component by network. The *E.coli* PPI network from DIP has 1440 nodes and 5871 edges; the *H.pylori* PPI network has 686 nodes and 1351 edges.

It is important to note that in order to determine the pathogenic proteins that have human orthologs, different methods can be used. Here we searched for each pathogenic protein among human proteins using BLASTp [1] with maximum E-value set to $10^{-5}$ and those pathogenic proteins which had at least 70% similarity with a human protein were considered to have human orthologs. Then proteins with human orthologs were assigned a very high weight to ensure that they would not be picked as potential drug targets.

Choosing essential pathways and complexes of interest and end node pairs within a pathway or complex. Given a number of essential pathways/complexes, it is of key importance to determine the end node pairs of interest from each pathway/complex that we want to disrupt. In this study, for any given pathway/complex we consider every possible pair of its end nodes; a pair of proteins in a pathway/complex are considered to be end nodes, if the number of hops between them is equal to the “diameter” (i.e. the maximum distance between two nodes) of the pathway/complex. Disrupting all the paths between such a pair in the PPI network will make sure that they will have no means of “communication” and thus the function associated with the pathway/complex will be disrupted as well.

It is also of importance which of the available pathways/complexes of a pathogen is of interest for drug target determination. In this study we have used essential pathways/complexes provided by public databases, in particular the KEGG database [19]. Note that there are also methods in the literature to determine “potential” pathways by the use of computational pathway/complex discovery tools such as [21] or [35]. Such methods were not used in this study.

In our experiments, we used two different sets of pathways and complexes on *E.coli*. One set was chosen from available pathways and complexes (excluding metabolic pathways) in the KEGG database. Another set we used was the essential and highly conserved protein complex data provided by [4], which studies a total of five such complexes. In our experiments on *H.pylori*, we used the set of pathways and complexes from KEGG database. The KEGG database provides both metabolic and signaling pathways and complexes for *E.coli* and *H.pylori*. This study focuses on all available signaling pathways of both *E.coli*, as listed in table 1 and *H.pylori* as listed in table 3. We picked all pairs of end nodes from each pathway/complex provided that both end nodes are present in the PPI network considered.
3.2 Discovering potential targets on known essential signaling pathways and complexes

We first discuss our results for the case that a number of essential pathways/complexes that can be targeted are available for a given pathogen.

**E.coli**  The KEGG [19] database provides 9 pairs of proteins that can be used as end nodes of *E.coli* signaling pathways; these end nodes are listed in table 1. In one experiment, we used this list as the set of input pathways/complexes to guide our combinatorial sparsest cut algorithm (DSC in short, see section Methods) as well as the algorithm based on linear programming (LP in short, see section Methods). In another experiment, we used 5 essential conserved complexes listed in [4] which provides, altogether, 7 pairs of end nodes given in table 2. In both experiments, we set the essentiality of each end node pair to 1 (\(\text{ess}(s_i, t_i) = 1\)). Both experiments were based on the the main connected component of the *E.coli* PPI network from the DIP database [34, 40] with 1440 proteins and 5871 interactions.

In the first experiment (on end node pairs from table 1), the DSC algorithm returned three proteins, whose removal completely disconnects three of the input end node pairs in the PPI network and thus disrupts the associated signaling pathways. It is important to note that two of these three proteins are known to be “essential proteins” [41, 12] (i.e. their removal is likely to be highly lethal to *E.coli*). Out of roughly 5000 *E.coli* proteins investigated, the total number of essential proteins identified so far is 400 [12]. The LP approach, on the other hand, identified two proteins whose removal disconnects a total of two end node pairs and their associated signaling pathways. The disrupted signaling pathways, the associated end node pairs disconnected and the proteins targeted by the two algorithms is shown in table 1.

For the list of protein pairs chosen from complexes listed in [4], both DSC and LP algorithms returned the same set of eleven proteins whose removal would disconnect/disrupt two of the input complexes (see table 2). Among these eleven proteins, two of them are well known drug targets, against which FDA approved drugs from the drugbank database [39] are currently in use. Note that among the 1440 proteins in the *E.coli* PPI network, only 15 of them are known drug targets. Furthermore, among the eleven proteins identified in this experiment, nine are known as essential proteins [12].

**Remark:** We note that it is possible to use our general approach for target identification through the use of MLC algorithm to discover novel drug targets which would maximize the lethality of available drugs that target specific pathways/complexes. This can be done by simply removing the targets of such drugs from the PPI network and use MLCalgorithm to obtain a set of proteins whose removal would guarantee the separation of source/sink pairs related to the pathways/complexes to be disrupted. One can apply this strategy to the RNA Polymerase complex (the main complex targeted by the available set of drugs against *E.coli*), which includes two proteins, rpoA and rpoB (targeted by conventional drugs Rifabutin, Rifampin and Rifaximin). The MLC algorithm applied to the *E.coli* PPI network after rpoA and rpoB are removed returns four additional potential drug targets, rpoC, rplC, rpsB and rpsE, among which only rpoC is in the complex. Proteins rplC, rpsB and rpsE are members of ribosomal complexes and cannot be considered as a backup for the polymerase complex. Thus, in the pruning step we remove these three proteins from target set.

**H.pylori**  We were able to find a total of 12 pairs of proteins from KEGG [19] database that can be used as end nodes of *H.pylori* signaling pathways; these end nodes are listed in table 3. We used
this list as the set of input pathways/complexes to guide our DSC and LP algorithms on the PPI network of *H. pylori* [34, 40] with 686 proteins and 1351 interactions. As in our experiments for *E. coli*, we set the essentiality of each end node pair to 1.

The results of both the DSC and the LP algorithms on this data set turned out to be identical: they returned a set of eight nodes whose removal disconnects exactly eight pairs of end nodes and their associated pathways. The disconnected node pairs, their associated pathways and target proteins are given in table 3.

### 3.3 Discovering potential targets without the guidance of known pathways

We used our HMWS heuristic (see subsection Minimum Weighted Node Separator) for partitioning both the *E. coli* and the *H. pylori* PPI networks. We compared our results with those obtained by two alternative heuristics: (1) a greedy heuristic which iteratively removes the highest degree node until a “balanced” cut is obtained (we would refer to this heuristic as GDeg), (2) another one which iteratively removes the node with the highest betweenness value until a balanced cut is obtained (we would refer to this heuristic as GBet).

Figure 2 shows how well these heuristics work on *E. coli* and *H. pylori* PPI networks in comparison to HMWS. As can be seen, our method always performs better than the alternatives for all values of the balance factor *β*, i.e. the number of targets it returns is always fewer than that obtained by the alternatives.

#### *E. coli*

As can be seen in figure 2 the number of targets returned by our approach varies significantly with the balance factor *β* used. Clearly those values of *β* for which we obtain too many targets are not of interest. However it is also of importance to identify the values of *β* and the number of targets identified respectively, whose removal disrupts several known essential pathways/complexes.

In table 4 we give the list of essential pathways/complexes which are disrupted by removing 28 targets identified by our method; this is achieved for *β* = 0.15 (HMWS with *β* = 0.15 found a cut of size 28 which disconnected 215 nodes from *E. coli* network.). We also look at the essential pathways/complexes disrupted by removing the same number of proteins (28) identified by the two alternative heuristics (obviously using different values of *β*). GDeg and GBet were just able to disrupt respectively the first 8 and 11 pathways listed in table 4. Many of these pathways are metabolic pathways, however, a few are signaling pathways. Those signaling pathways for which all possible backup paths are also disrupted by the removal of the suggested targets are marked with a † in table 4 (Note that only the proposed heuristic HMWS was able to disrupt these 4 pathways and all of their possible backup paths).

Note that as it is shown in table 5, 4 proteins among the 28 identified as a cut by HMWS with *β* = 0.15, are well known drug targets for *E. coli* [39]. (there are only total of 15 known drug targets in drugbank for which FDA approved drugs exist.)

One further experiment we conducted aims to see how HMWS algorithm can help identify proteins that can be targeted so as to improve the effectiveness of known drugs and their targets. For this purpose, we add the set of predefined targets to the initial cut picked in the first step (*split step*) of the HMWS algorithm (so as to significantly increasing their chances of ending up in the final cut), and then continue with the algorithm as before. In table 6 we demonstrate the size of cut for varying *β* values, when all of *E. coli*'s 15 known drug targets are picked first in the split step. We compare the results to the original application to the HMWS algorithm where no such biases is imposed. Interestingly, the size of the cut decreases when we are guided with the known
drug targets for $\beta$ values 0.20 and 0.25. The potential targets identified included at least 8 of the 15 known targets.

In final experiment we aim to further investigate the potential of formulating the target discovery problem as a minimum weighted node separator problem. Note that HMWS is only a heuristic which seems to work well in practice but is not guaranteed to find the optimal solution to the minimum weighted node separator problem. It is possible that all (or a significant majority of) known drug targets are a part of an optimal solution to this problem; if they are, it would imply the validity of the general approach and the need for developing a better tool for computing the optimal solution.

In this new experiment we aimed to find a cut that includes all 15 known drug targets [39] in E.coli and achieves $\beta = 0.25$. We compared this cut to one obtained by the direct use of HMWS, again with $\beta = 0.25$. Thus, we first manually removed the 15 known drug targets from the E.coli PPI network and then applied HMWS on the remaining portion of the PPI network to find a minimum weighted node separator with $\beta = 0.25$. We were able find a cut of size 44 on this restricted network, which implies a total cut size of 59 for the complete PPI network. The original cut size obtained by the HMWS algorithm with identical $\beta = 0.25$ was only able to find a cut of size 60 (in this cut only 6 known drug targets were included). The above experiment illustrates that there are cuts which are smaller in size to the one found by HMWS and include all known drug targets. Thus the general approach we propose here could in fact provide more interesting solutions if the computational method employed for solving the minimum weighted node separator problem could be improved. We believe that approximation algorithms based on semi definite programming [8, 2] may be able to provide such improvements; although existing implementations of these algorithms are too slow to be practical, we may be able to use them for identifying interesting potential drug targets in the near future.

**H.pylori** The final experiment we conducted was on H.pylori PPI network using HMWS with $\beta = 0.15$ which returned 17 proteins as potential targets. In table 7 we list the essential pathways/complexes from KEGG [19] database which are disrupted by removing these 17 targets. Interestingly, as it is shown in table 7, the removal of this target set disrupts a significant number of signaling pathways together with all possible backup paths in PPI network (the signaling pathways/complexes for which all backup paths are also disrupted are marked with †). Note that one of the targets chosen by HMWS when $\beta = 0.15$ is a well known vaccine target named vacuolating cytotoxin VacA [16]).

4 Conclusion

In this study, we present two general strategies for identifying potential multiple drug targets in pathogens based on computational exploration of their PPI networks. The first strategy aims to disrupt available essential pathways and complexes as well as all possible backup paths in PPI networks; it typically returns a small (and thus practical) set of potential drug targets. We developed and tested two algorithms for this purpose; DSC and LP. In the E.coli PPI network, given 9 end pairs of pathways (chosen as described in section Results from all available signaling pathways from KEGG [19]), DSC returned 3 target proteins to disrupt 3 of these pathways and LP returned 2 proteins to disrupt 2 of these pathways; both of results maximize the ratio of the number of disrupted pathways to the number of selected targets as we had hoped.

The majority of proteins returned as drug targets by LP and DSC were essential proteins, as
given in [41]. In addition to this, the same experiments were performed referring to end pairs of essential complexes (see again [4]), rather than pathways, of E.coli. LP and DSC both returned a set of 12 proteins each whose removal disconnects 2 of these pairs. Among these 12 proteins, 10 are essential; even more importantly 2 of them are known drug targets. These observations seem to support that aiming to maximize the ratio of the number of disrupted pathways to the number of drug targets gives a small and practical set of drug targets.

The second strategy was developed for the case where there is limited or no information about essential pathways/complexes in a pathogen. As per the first one, the second strategy aims to maximize the number of disrupted “potential” pathways/complexes by removing the minimal number of selected targets. More specifically, our heuristic approach that employs this strategy, HMWS, partitions the PPI network into two almost-equal sized subnetworks so as to maximize the number of potential pathways/complexes disrupted. We compared HMWS method against two simpler heuristics such as GDeg (one which removes the highest degree vertex iteratively until a cut is obtained) and GBet (one which removes the vertex with highest betweenness). The HMWS clearly outperforms both heuristics on all data sets we used. Here, sizes of sets of potential targets (28 in E.coli and 17 in H.pylori) are larger and, possibly, need to be pruned further. Again, the majority of proteins returned by the HMWS are essential [41]; in addition, a fair portion of them are known drug targets. In sum, we have demonstrated that, through the second strategy, removing minimal sets of proteins that disrupt the cellular organizations of the pathogens yields small and effective sets of potential drug targets.

Note that in this study we have excluded the metabolic pathways: the primary reason is that metabolic pathways involve chemical compounds, which are not represented in PPI networks. As a result the effect of targeting proteins in a metabolic pathway towards disrupting its function is not very clear even when one can guarantee that all backup paths have been disrupted. It is possible to extend our methods so that the KEGG [19] representation of metabolic pathways (which is ordinary graph where nodes are proteins or chemical compounds) are considered as well. For that, we need to extend the PPI network, to one which not only includes all interactions between proteins, but also between proteins and chemical compounds. In order to make sure that none of the chemical compounds will be chosen, we have to assign large weights to the nodes that represent chemical compounds.

The two strategies investigated in this paper are quite fast and accurate. It remains an open question whether the improved solutions can be obtained by novel alternative approximation algorithms or heuristics for solving well known combinatorial problems to which the drug target discovery problem can be reduced. The reliability of our results will also improve as more complete PPI networks become available for pathogens investigated in this study.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. Journal of Molecular Biology, 215(403-410), 1990.

[2] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pages 227–236, 2007.

[3] A. A. Borisy, P. J. Elliott, M. S. Lee N.W. Hurst, J Lehar, ER Price, G Serbedzija, GR Zimmermann, MA Foley, BR Stockwell, and CT Keith. Systematic discovery of multicomponent therapeutics. Proceeding of National Academical Science, 100:7977–7982, 2003.
[4] Gareth Butland, Joe Manel Peregrn-Alvarez, Joyce Li, Wehong Yang, Xiaochun Yang, Veronica Canadien, Andrei Starostine, Dawn Richards, Bryan Beattie, Nevan Krogan, Michael Davey, John Parkinson, Jack Greenblatt, and Andrew Emili. Interaction network containing conserved and essential protein complexes in escherichia coli. *Nature*, 433:531–537, 2005.

[5] Alison F. Chalker, Heather W. Minehart, Nicky J. Hughes, Kristin K. Koretke, Michael A. Lonetto, Kerry K. Brinkman, Patrick V. Warren, Andrei Lupas, Michael J. Stanhope, James R. Brown, and Paul S. Hoffman. Systematic identification of selective essential genes in helicobacter pylori by genome prioritization and allelic replacement mutagenesis. *Journal of Bacteriology*, 183(4):1259–1268, 2001.

[6] P. Csermely, V.´Agoston, and S. Pongor. The efficiency of multi-target drugs: the network approach might help drug design. *Trends in Pharmacological Sciences*, 26(4):178–182, 2005.

[7] Ernesto Estrada. Protein bipartivity and essentiality in the yeast protein-protein interaction network. *Journal of Proteome Research*, 5(9):2177–2184, 2006.

[8] Uriel Feige, MohammadTaghi Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum-weight vertex separators. In *STOC ’05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 563–572, 2005.

[9] S. Frantz. Drug discovery: Playing dirty. *Nature*, 437:942–943, 2005.

[10] L. C. Freeman. A set of measuring centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[11] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(4):237–267, 1976.

[12] S. Y. Gerdes, M. D. Scholle, J. W. Campbell, G. Balzsi, E. Ravasz, A. L. Somera, N. C. Kyripides, I. Anderson, M. S. Gelfand, A. Bhattacharyya, V. Kapralai, M. D’Souza, M. V. Baev, Y. Grechkin, F. Mseeh, M. Y. Fonstein, R. Overbeek, A. L. Barabasi nad Z. N. Oltvai, and A. L. Osterman. Experimental determination and system level analysis of essential genes in escherichia coli mg1655. *Journal of Bacteriology*, 185(19):5673–5684, 2003.

[13] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of ACM*, 35(4):921–940, 1988.

[14] M. T. Hajiaghayi and H. R¨acke. An $O(\sqrt{\tau})$-approximation algorithm for directed sparsest cut. *Information Processing Letters*, 97(4):156–160, 2006.

[15] S. Hasan, S. Daugelat, S. S. Rao, and M. Schreiber. Prioritizing genomic drug targets in pathogens: Application to mycobacterium tuberculosis. *PLoS Computational Biology*, 2(6):e61, 2006.

[16] C. Hatzifoti, B. W. Wren, and W. J. Morrow. Helicobacter pylori vaccine strategies–triggering a gut reaction. *Immunology Today*, 21(12):615–619, 2000.

[17] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 65(5 Pt 2), 2002.

[18] S. Huang. Rational drug discovery: what can we learn from regulatory networks? *Drug Discovery Today*, 7(20):163–169, 2002.

[19] M Kanelhisa, S Goto, M Hattori, KF Aoki-Kinoshita, M Itoh, S Kawashima, T Katayama, M Araki, and M Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(Database issue):D354–357, 2006.

[20] D. B. Kell. Systems biology, metabolic modelling and metabolomics in drug discovery and development. *Drug Discovery Today*, 11(23-24):1085–1092, 2006.

[21] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*, 100(20):11394–11399, 2003.

[22] T. Korcsmaros, M. S. Szalay, C. Bode, I. A. Kovacs, and P. Csermely. How to design multi-target drugs: Target search options in cellular networks. *Expert Opinion on Drug Discovery*, 2(6):799–808, 2007.

[23] Tom Leighton, Fililia Makedon, Serge Plotkin, Clifford Stein, Eva Tardos, and Spyros Tragoudas. Fast approximation algorithms for multicommodity flow problems. *Journal of Computer and System Sciences*, 50(2):228–243, 1995.

[24] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of ACM*, 46(6):787–832, 1999.

[25] Robin Lougee-Heimer. The common optimization interface for operations research. *IBM Journal of Research and Development*, 47(1):57–66, 2003.
[26] D. W. Matula and F. Shahrokhi. The maximum concurrent flow problem and sparsest cuts. *Technical Report Southern Methodist University*, 1986.

[27] W. G. Kaelin Jr. Gleevec: Prototype or outlier? *Science STKE*, 2004(225):pe12, 2004.

[28] World Health Organization. *The world health report 2007 - A safer future: global public health security in the 21st century*. WHO, 2007.

[29] M. J. Millan. Multi-target strategies for the improved treatment of depressive states: Conceptual foundations and neuronal substrates, drug discovery and therapeutic application. *Pharmacol. Therap.*, 110:335–370, 2006.

[30] R. Morphy, C. Kay, and Z. Rankovic. From magic bullets to designed multiple ligands. *Drug Discovery Today*, 9(15):641–651, 2004.

[31] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

[32] N. Pržulj, D. A. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics*, 20(3):340–348, 2002.

[33] B. Przydatek. A fast approximation algorithm for the subset-sum problem. *International Transactions in Operational Research*, 9(4):437–459, 2002.

[34] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(Database issue):D449–451, 2004.

[35] J. Scott, T. Ideker, R. M. Karp, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, 13(2):133–144, 2006.

[36] J. R. Sharom, D. S. Bellows, and M. Tyers. From large networks to small molecules. *Current Opinion in Chemical Biology*, 8(1):81–90, 2004.

[37] M. Strong and D. Eisenberg. The protein network as a tool for finding novel drug targets. *Progress in Drug Research*, 64:191–215, 2007.

[38] G. C. Terstappen and A. Reggiani. In silico research in drug discovery. *Trends in Pharmacological Sciences*, 22(1):23–26, 2001.

[39] David S. Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, 34(Database issue):D668–672, 2006.

[40] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. Kim, and D. Eisenberg. Dip, the database of interacting proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305, 2002.

[41] Ren Zhang, Hong-Yu Ou, and Chun-Ting Zhang. Deg, a database of essential genes. *Nucleic Acids Research*, 32:D271–D272, 2004.

[42] G. R. Zimmermann, J. Lehar, and C. T. Keith. Multi-target therapeutics: when the whole is greater than the sum of the parts. *Drug Discovery Today*, 12(1-2):34–42, 2007.

5 Tables
Table 1: Pairs of end node proteins from *E. coli* signaling pathways from the KEGG database [19]. Also shown is the target proteins which would disrupt the pathway (essential proteins are marked with *) and the methods that identified each protein.

| Pathway                  | “Source” | “Sink” | Target(s) | Method(s) |
|--------------------------|----------|--------|-----------|-----------|
| OmpR Family              | PhoR     | PhoA   |           |           |
| OmpR Family              | TorS     | TorA   |           |           |
| NarL Family              | NarG     | NarI   |           |           |
| NarL Family              | NarQ     | FruA   | dnaK*     | DSC , LP  |
| DNA Polymerase           | dnaE     | holA   | holD      |           |
| ABC Transporter          | CysP     | CysA   |           |           |
| Bacterial Chemotaxis     | tar      | MotA   | cheW*     | DSC , LP  |
| Bacterial Chemotaxis     | DPPA     | MotA   | mtlD      | DSC       |

Table 2: Pairs of end node proteins chosen from conserved essential complexes in *E.coli* [4]. Also shown is the target proteins which would disrupt the complex (essential proteins are marked with * and known drug targets are marked with +) and the methods that identified each protein.

| Complex                | “Source” | “Sink” | Target(s) | Method(s) |
|------------------------|----------|--------|-----------|-----------|
| RNA Polymerase         | infB     | rpoN   |           |           |
| RNA Polymerase         | hepA     | greB   | rpoA*+, rpoB*+, rpoC*+, rpsB*+, rpsE* | DSC , LP  |
| Acetyl-CoA Synthase     | f6h      | aidB   |           |           |
| LysS                    | lscA     | fdiD   | lpdA*, lysU, aceF*, aceE, iscS*, rpsE* | DSC , LP  |
| DNA Polymerase          | sscB     | priA   |           |           |
| Ribosome associated     | hlpA     | uvrC   |           |           |
| Ribosome associated     | cafA     | ldsA   |           |           |

Table 3: Protein pairs chosen from the signaling pathways of *H. pylori* from the KEGG database [19]. Also shown is the target proteins which would disrupt the pathway and the methods that identified each target.

| Pathway                  | “Source” | “Sink” | Target(s) | Method(s) |
|--------------------------|----------|--------|-----------|-----------|
| Ribosomal Proteins       | rplD     | rplP   |           |           |
| Ribosomal Proteins       | rplI     | rplF   |           |           |
| Type III Secretion Sys.  | FliF     | FliA   |           |           |
| Type IV Secretion Sys.   | cag12    | trbI   |           |           |
| Two Component Sys.       | TrpB     | TrpE   |           |           |
| Flagellar Assembly       | FliG     | FliN   |           |           |
| Bacterial Chemotaxis     | CheW     | MotB   |           |           |
| ABC Transporters         | OppA     | OppP   |           |           |
| DNA Polymerase           | dnaE     | dnaN   |           |           |
Table 4: List of disrupted *E.coli* pathways by various methods with cut size 28 ($\beta = 0.15$). Marked with † are those pathways for which all backup paths in the PPI network are also disrupted. With the same cut size GDeg and GBet were just able to disrupt respectively the first 8 and 11 listed pathways.

| Disrupted Pathways |
|---------------------|
| 1 Ribosome          |
| 2 Pyruvate metabolism|
| 3 Butanoate metabolism|
| 4 Citrate cycle (TCA cycle) |
| 5 Glycolysis/Gluconeogenesis |
| 6 Alanine and aspartate metabolism |
| 7 Glycine, serine and threonine metabolism |
| 8 Valine, leucine and isoleucine degradation |
| 9 Pyrimidine metabolism |
| 10 Purine metabolism |
| 11 RNA polymerase |
| 12 Lysine biosynthesis |
| 13 Aminoacyl-tRNA biosynthesis |
| 14 Two component (NarL family)† |
| 15 Bacterial Chemotaxis† |
| 16 ABC transporters (Iron complex)† |

Table 5: Known *E.coli* drug targets in a cut of size 28 chosen by HMWS method ($\beta = 0.15$).

| Gene Name | Drug |
|-----------|------|
| rpoA      | Rifabutin |
| rpoB      | Rifampin, Rifaximin |
| rpsJ      | Nitrofurantoin |
| rpsD      | Clomocycline, Demeclocycline, Doxycycline, Lymecycline, Minocycline, Oxytetracycline, Tetracycline, Tigecycline |

Table 6: The cut size for different values of $\beta$ when guided by 15 known drug targets to the case that no list of known drug targets is available. The + sign indicated the known drug targets.

| HMWS guided by known targets | HMWS (original) |
|------------------------------|------------------|
| Cut Size | Number of Approved Targets | Cut Size | Number of Approved Targets |
| $\beta = 0.10$ | 21 $^a$ | 8 | 18 $^a$ | 3 |
| $\beta = 0.15$ | 31 | 9 | 28 | 4 |
| $\beta = 0.20$ | 40 | 9 | 42 | 5 |
| $\beta = 0.25$ | 55 | 9 | 60 | 6 |

$^a$Proteins in cut are groL, lpdA, rpoA+, rpoB+, rplL, murB+, rpsB, tufB, aceE, dnaK, P20082+, rplV, polA, yfgB, rpoC, rpsJ+, tufA, rpsE, ddlA+, gyrB+, frdA+.
$^b$Proteins in cut are lpdA, rpoA, rpoB, rplL, lpsU, aceE, dnaK, aceF, rplC, rplV, rpoC, groL, rpsJ, secA, tufA, rpsE, yfgB.
Table 7: List of disrupted *H. pylori* pathways by HMWS with cut size 17 (\(\beta = 0.15\)). Marked with † are those pathways for which all backup paths in the PPI network are also disrupted.

| Disrupted Pathways                                      |
|---------------------------------------------------------|
| 1. Purine metabolism                                    |
| 2. Pyrimidine metabolism                                |
| 3. RNA polymerase                                        |
| 4. Caprolactam degradation                               |
| 5. RNA polymerase†                                       |
| 6. Urease complex                                        |
| 7. Ribosomal proteins†                                   |
| 8. Oxidative phosphorylation (F-type ATPase)†            |
| 9. Epithelial cell signaling in *H. pylori* infection†   |
| 10. DNA polymerase†                                      |
| 11. Bacterial chemotaxis†                                 |
| 12. Oxidative phosphorylation (F-type ATPase)†           |
| 13. Protein export (Sec dependent pathway)†              |
| 14. ABC transporters (Iron complex)†                     |
| 15. Two-component system - NtrC family†                  |
| 16. Flagellar assembly†                                  |
| 17. Type IV secretion system †                           |

(a) *H. pylori* Chemotaxis pathway.
(b) *H. pylori* PPI subnetwork.

Figure 1: Chemotaxis pathway and associated PPI subnetwork of *H. pylori*. 
Figure 2: Comparing the efficiency of HMWS with alternative heuristics GDeg and GBet. The plot shows how the cut size changes with respect to the balance factor $\beta$. 

(a) *E.coli* 

(b) *H.pylori*