Network controllability solutions for computational drug repurposing using genetic algorithms

Victor-Bogdan Popescu

Abo Akademi University

Krishna Kanhaiya

Abo Akademi University

Iulian Nastac

Politehnica University of Bucharest

Eugen Czeizler

Abo Akademi University

Ion Petre (✉ ion.petre@utu.fi)

University of Turku

Research Article

Keywords: cascading signals, bio-medical dynamical networks, signalling networks, immunological responses, drug targets

Posted Date: August 6th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-778595/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Scientific Reports on January 26th, 2022.
See the published version at <https://doi.org/10.1038/s41598-022-05335-3>.

Network controllability solutions for computational drug repurposing using genetic algorithms

Victor-Bogdan Popescu¹, Krishna Kanhaiya¹, Iulian Năstac², Eugen Czeizler^{1,4}, and Ion Petre^{3,4,*}

¹Abo Akademi University, Computer Science, Turku, 20500, Finland

²Politehnica University of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, Bucharest, 061071, Romania

³University of Turku, Department of Mathematics and Statistics, Turku, 20014, Finland

⁴National Institute for Research and Development in Biological Sciences, Bucharest, 060031, Romania

*Correspondence to: ion.petre@utu.fi

ABSTRACT

Control theory has seen recently impactful applications in network science, especially in connections with applications in network medicine. A key topic of research is that of finding minimal external interventions that offer control over the dynamics of a given network, a problem known as network controllability. We propose in this article a new solution for this problem based on genetic algorithms. We tailor our solution for applications in computational drug repurposing, seeking to maximize its use of FDA-approved drug targets in a given disease-specific protein-protein interaction network. We show how our algorithm identifies a number of potentially efficient drugs for breast, ovarian, and pancreatic cancer. We demonstrate our algorithm on several benchmark networks from cancer medicine, social networks, electronic circuits, and several random networks with their edges distributed according to the Erdős-Rényi, the scale-free, and the small world properties. Overall, we show that our new algorithm is more efficient in identifying relevant drug targets in a disease network, advancing the computational solutions needed for new therapeutic and drug repurposing approaches.

Introduction

Network modeling in systems medicine has emerged as a powerful analytics approach in the last couple of decades^{1,2}. Its aim is to analyze diseases and drug interventions as ways of acting over bio-medical dynamical networks^{3,4}, such as the protein-protein interaction networks^{5,6}, signalling networks⁷, metabolic networks⁸, and immunological responses⁹. In this framework, a disease is seen as emerging from some of its modules being affected (directly or through cascading signals) and from critical nodes in the network being deregulated¹⁰. Similarly, drug therapies are seen as outside controlled interventions within a deregulated network with the aim of either re-balancing the system or possibly isolating some specific components of the network¹¹. A particular advantage of this approach is reasoning about multiple-drug interventions, analyzing and predicting multi-drug synergies. Instead of acting over each individual dysregulated component, one can try to influence several of these entities through a few well-chosen interventions, and to have them spread in cascade into the network using the network's own internal interconnections. Network controllability is a topic of high relevance in this area with a rich theory to support it¹². It has found in recent years powerful applications in computational systems medicine and therapeutics^{5,6,13,14}.

The theory of network controllability aims at providing sound and theoretically accurate description of what control means within a network, and how it can be achieved. Intuitively, achieving control over a system from a set of input nodes means being able to drive that system from any initial setup to any desired state. This is an intrinsic optimization problem with the objective to minimize the number of input nodes (e.g., drug targets) needed for the control. Additional constraints may be added depending on the application, such as the control pathways from the input nodes to the controlled nodes to be short, or the input nodes to be primarily selected from a given set of preferred nodes (targets of standard therapy drugs). This leads to several problem variations, such as: structural controllability¹³, i.e., identifying pathways that offer control over the system regardless of its numerical setup; target controllability¹⁴, i.e., achieving control over a predefined set of target nodes; minimum dominating sets¹⁵, i.e., finding a minimal set of nodes that are one step upstream of all other nodes in the network. Some of these optimization problems are known to have efficient algorithmic solutions¹³. Others, on the contrary, are known to be computationally difficult, yet approximate efficient solutions are still achievable⁵.

Motivated by the applicability of network control in systems medicine, the problem we focus on in this paper is minimizing

the number of external interventions needed to achieve structural target control of a system. We are particularly interested in the case where the targets are disease-specific survivability-essential genes, key targets for synthetic lethality¹⁶. We identify control interventions that are achievable through the delivery of FDA-approved drugs, by giving a preference to FDA-approved drug targets being selected as input nodes. The minimization of the number of input nodes is part of the standard optimization objective of the network controllability problem and there is an additional incentive to minimize them in the medical case studies: combinatorial drug therapies can only include few simultaneously delivered drugs. The optimization goal of the network controllability problem can address this constraint by selecting the input nodes from among the drug-approved nodes.

The structural target controllability problem is known to be NP-hard, meaning that finding the smallest set of inputs for controlling the target set is computationally prohibitive for large networks. We give as a solution an approximation of the minimal solution based on genetic algorithms, well known heuristic choices for nonlinear optimization problems¹⁷. We demonstrate that this approach offers an efficient solution for applications in combinatorial drug therapy identification and drug repurposing.

Results

Comparison setup

We compared the results of our genetic algorithm to the results of the greedy algorithm for the structural target controllability described in⁵, applied on the same data sets. In the cases where we considered preferred input nodes, both algorithms were optimized to select a maximal number of such nodes in their solutions (the optimized greedy algorithm is detailed in⁶).

The search for solutions is completely different in the genetic and in the greedy algorithm. The greedy algorithm finds the solutions by constructing the control pathways step by step and elongating them as long as it takes to minimize the number of nodes they start from (which will become the input nodes in the solution). This way, they output the control pathways as a solution, at the cost of potentially obtaining impractically long control pathways. In contrast, the genetic algorithm skips the search for the pathway and only focuses on finding the input nodes offering control over the target nodes. Additionally, they only search at a set distance upstream of the target nodes, as specified in the parameter $\max_{path}$. The input nodes are verified to form a control set for the target nodes through the Kalman rank condition. In this way, the genetic algorithm offers control sets, but not the control pathways from the input nodes to the target nodes and, consequently, it does not report on the target nodes controlled from each input node. One can calculate the strength of the control offered by each set of input nodes alone: the rank of the Kalman matrix with the input set consisting of a single node only is exactly the number of target nodes that are possible to control from that node.

The algorithms are quite different also in their output procedure. The genetic algorithm maintains in each generation of its search a family of solutions, some better than others (in terms of the number of distinct genes on each chromosome), but all valid solutions for controlling the given set of targets. At the end of each run of the genetic algorithm we output the solutions of minimum size (i.e., all chromosomes with minimal number of genes), potentially offering several different solutions (of equal size). This is in contrast with the greedy algorithm, where only one solution is found in one run and multiple runs have to be done to collect multiple solutions (potentially of different size in different runs).

To investigate the effect of the limited length pathways in our algorithm, we ran the greedy algorithm in two different settings: with the control path's length bounded by the same parameter as in the genetic case, and with it unconstrained. All runs that didn't complete within three days were stopped.

Benchmark data

We applied the algorithms on several real world and randomly generated complex networks. The size of the networks varied from 32 to over 3000 nodes. An overview of the data sets is presented in the supplementary table 1. We used the 4 breast, pancreatic, and ovarian cancer cell line-specific protein-protein interaction networks documented in⁶. We eliminated all edge duplicates and target node duplicates from those networks, as well as all isolated nodes. We also used the 6 breast, pancreatic, and ovarian cancer networks of¹⁸. Additionally, we considered the 2 social interaction networks and 3 electronic circuit networks documented in¹⁹ and²⁰. In the social networks the nodes represent people in a group, and the edges represent positive sentiments from one person towards another. In the electronic circuit networks the nodes represent logic gates and the edges represent the connections between them. Finally, we generated 21 random graphs with nodes ranging from 100 to 3000 and the edges distributed according to the Erdős-Rényi-, scale-free-, and small world-graph edge distributions, all of them generated using the Python *networkx* library²¹.

For the protein-protein interaction networks we used as target nodes the cancer essential genes specific to each cell line, based on¹⁸. For the other networks, as target nodes we chose the top 5% nodes with highest degree and, in a separate series of tests, sets consisting of 5% randomly selected nodes with positive in-degree (so they have a chance to be controlled). In addition, we applied again the algorithm on the protein-protein interaction networks considering as preferred nodes the FDA-approved drug-targets²² in each network.

All runs were executed with the same default values for the parameters of the algorithm (except for max_{path}), as detailed in table 1. The max_{path} parameter indicates the maximum length of a control path in our solution. For the biological networks we set the value of this parameter to 5 to search for results with plausible applicability. For the non-biological networks we set the value of this parameter to 50 to compare the scalability of the three methods.

All of the data and algorithm results are available in the application repository at²³.

Generating the networks

The set of interactions included in the network being analyzed is of crucial importance when applying network controllability for drug repurposing. As a proof-of-concept demonstration of the applicability of our algorithm, we chose in this study the cancer interaction networks from⁶. These networks were constructed based on data from the UniprotKB protein database²⁴, with the interactions coming from the SIGNOR database²⁵. Disease-specific markers and proteins were selected from UniProtKB and interactions between them (also through 2 intermediary additional, non-disease-specific proteins) were added based on SIGNOR.

The approach we took in this study is based on unlabelled networks, i.e., where we only include the information that a certain directed interaction between two proteins exists, without the details on the nature (e.g., inhibiting or activating) or the strength of that interaction. This is indeed the most common approach to network controllability, that may lead to false positive results regarding the therapeutic effects: some of the control pathways we identify may have a weak result due to the conflicting contributions of the interactions it consists of. A modest amount of false positives is sometimes tolerated in medical research, more than false negatives, because it leads to a wider pool of candidates to be verified experimentally. Approaches representing the type of interaction within the network and within the controllability problem do exist, e.g. based on Boolean networks^{26,27}, but they are hindered by major scalability problems.

Comparison results

The first benchmark objective we compared against was the size of the smallest set of input nodes found by each of the three algorithms, with the smallest being the best. To make the comparison possible, we ran each algorithm once for each of the benchmark networks, until there was no improvement in the size of the best result for 1,000 iterations (translated to 1,000 generations for the genetic algorithm). To ensure an upper bound on the processing time, we also encoded a maximum hard limit of 10,000 iterations/generations, which, however, was never reached in any of the runs. The results are presented in the supplementary table 1, in the supplementary figure 1, and they are discussed below.

Next, we compared the ability of the three algorithms to maximize the use of preferred nodes. We did this on the biological networks, using as preferred nodes the set of FDA-approved drug targets. We also did a literature-based validation of the relevance of the results found by the genetic algorithm in the cancer networks. The results are presented in the supplementary tables 2, 3, 4, in the supplementary figure 2, and they are discussed below.

Additionally, in order to compare the running times per computed solution for each algorithm, we selected three similar-sized random graphs and we ran the algorithms iteratively for 300 times on each of them. Each iteration of the greedy algorithm consisted of 80 runs, offering 80 solutions per iteration. This corresponds to the genetic algorithm maintaining a population of 80 in each generation, thus constructing simultaneously 80 solutions per iteration. The results are presented in the supplementary figure 3 and discussed below.

Finally, to measure the algorithm convergence speeds, we ran the default versions of the algorithms iteratively for just 60 minutes. The results are discussed below.

Size of the best solutions

In the case of the cancer protein-protein interaction networks, the genetic algorithm returned input sets of size 10 – 25% smaller than the constrained greedy algorithm and 10 – 20% smaller than the general greedy algorithm. The comparison can be seen in the supplementary figure 1. In the case of the non-biological networks, the size of the smallest input sets is virtually identical in the three algorithms, even though the sets themselves may be different. We believe the reason for why the genetic algorithm managed to improve the results of the greedy algorithms in the case of the biological networks and only match them for the other networks is in the size of the target set and in the search strategy the algorithms adopt. The target set we worked with in the case of the biological networks is larger relative to the size of the network than in the case of the non-biological networks. Consequently, the set of predecessors of the target nodes, i.e., the potential input nodes, is larger in this case relative to the size of the network. The genetic algorithm sampled freely from this larger set to find potential input sets, whereas the greedy algorithms constructed control pathways step-by-step and as a result, constrained their set of potential input nodes in each step. The difference became visible in the case of the biological networks because of the larger set of potential input nodes.

An interesting aspect can be seen in the analyses of the well-connected Erdős-Rényi and small world random networks, where the genetic algorithm succeeds in finding the optimal solution. To see this, let's consider the case of the largest small world benchmark network. The maximum path length being set at 50 means that a given input node can control at most 51

nodes in the network (itself, and 50 others, through consecutive edges). Thus, for a set of 150 target nodes, there have to be at least $\lceil \frac{150}{51} \rceil + 1 = 3$ input nodes. The genetic algorithm indeed identifies exactly 3 input nodes, with a maximum length of the control path of 49. The general greedy algorithm, being allowed to use longer control paths, identified one input node controlling the entire target set, at the cost of an increased maximum path length of 149 and a running time 40 times longer.

Running times and convergence speed

Within the runs aimed at measuring the size of the best solutions, the genetic algorithm was the fastest in the random Erdős-Rényi networks, in the random small world networks, and in some of the real-life non-biological networks. For the biological and the scale-free networks the fastest was the constrained greedy algorithm, with the genetic algorithm the slowest of the three, and in general 2-4 times slower than the general greedy algorithm, as it can be seen in the supplementary figure 1. However, the genetic algorithm provides simultaneously several solutions (in our benchmarks tests we set the population size parameter to 80). This is a key advantage of this algorithm, especially for drug repurposing applications, where multiple alternative solutions are important to collect and compare. Thus, we performed a set of additional runs to investigate the running times per solution. As it can be seen in the supplementary figure 3, in the case of the random Erdős-Rényi network, the three algorithms have comparable running times per solution (as well as size of the obtained input sets). On the other hand, in the case of the sparser-connected scale-free network, the greedy algorithms outperformed the genetic one, however, at the cost of an increased size of the best solutions. Finally, in the case of the small world network, the genetic algorithm had a running time comparable to the constrained greedy (while consistently returning much smaller solutions), and 10 times faster than the general greedy algorithm. Nonetheless, another difference in the results was in the number and diversity of the identified solutions, where the genetic algorithm returns more than 2 times more unique solutions of minimum size compared to its greedy counterparts.

Therefore, the number of edges in a network alone does not seem to influence the performance of the algorithm. The increase in the running time as the network size increases is mainly caused by the increasing number of nodes and target nodes. This makes the genetic algorithm well suited to be applied on better connected networks.

In addition, we compared the evolution of the quality of the solutions throughout the generations for each of the additional runs and we noticed that a good solution (i.e., a solution with the fitness within 10% of the best final solution) is, in fact, achieved very quickly, typically within a few tens of generations from the start. Supplementary figure 4 illustrates the average and best fitness of the chromosomes in each generation of the algorithm over the additional runs for each of the three analyzed networks. This suggests that the genetic algorithm may be applied successfully with a much lower number of generations, perhaps as low as 100, adding a considerable speed-up to it. We checked and concluded that the situation is similar with the greedy algorithms: the best solution is typically obtained quickly, in the first few iterations.

Furthermore, most of the best solutions returned by the additional runs are of the minimum size (although the content of the actual solution may vary). This proves the consistency of the genetic algorithm, suggesting that only a small number of runs is required in order to achieve a best final result. This is confirmed by a final set of runs, aimed at checking the time elapsed for the algorithm to find its best solution for each of the networks. Indeed, in all of the analyzed cases, a solution of minimum size was found within the very first runs.

The three algorithms scaled up differently for different types of large networks. The unconstrained greedy algorithm did not finish within three days for the large Erdős-Rényi networks and partially for the large small world networks. On the contrary, the genetic algorithm did not finish within three days on the larger scale-free networks. We believe the difference is in the search strategy employed by each of these algorithms. The greedy algorithms extend the control pathways upwards with predecessors of their current starting nodes, aiming to minimize the number of predecessors selected in each step. This strategy becomes problematic in networks with a large number of edges: such networks tend to have many cycles, limiting the greedy algorithms' ability to decrease the number of starting points. The genetic algorithm looks ahead for predecessors at distance $\max_{path}$ from the control target nodes and aims to find such a set of predecessors leading to a Kalman matrix of full rank. This strategy becomes problematic in scale-free networks that have a small number of hubs: such hubs are likely to be selected as predecessors for multiple control target nodes, while each being able to control by itself only a few of them, leading to a Kalman matrix of non-full rank.

Maximizing the number of preferred nodes

We applied the three algorithms on the biological networks with the additional optimization objective of maximizing the selection as input nodes of FDA-approved drug targets in the network (preferred nodes). In all cases, the sets of input nodes returned by the genetic algorithm contained more preferred nodes than the ones returned by the other algorithms (supplementary figure 2), with the percentage of the preferred nodes relative to the size of the input nodes in the best solution being, in general two to four times higher in the case of the genetic algorithm. This led to more control target nodes being controlled by preferred nodes, i.e., leading to predictions of potentially more efficient drugs. This has as a consequence a clear improvement in the applicability of the algorithm in the biomedical domain for drug repurposing, an aspect that we discuss next.

Therapeutically-relevant findings

We analysed the FDA-approved drug targets predicted by our algorithm to control the highest number of essential genes in each of our network. The results are presented in the supplementary table 3. We used the public DrugBank database²² to find drugs targeting the proteins in the supplementary table 3 and known to be used in cancer therapeutics. The results are presented in the supplementary table 4.

For each cancer type we ran our algorithm on the corresponding networks (three for each cancer type, see the supplementary table 2). The algorithm offered for each network 80 sets of control input nodes, of which we selected those having the minimum number of elements, i.e., the optimal solutions for that network. The number of optimal solutions varied from network to network depending on the network topology and the results of the randomized choices in each run. To avoid spurious selections of input nodes, we list in the supplementary table 3 all input nodes having the average number of occurrences per solution at least two. To obtain the number of target nodes that each of these input nodes controls, we calculated for each input node i the rank of the Kalman matrix with the input set consisting of node i alone. The result is reported for each input node in the supplementary table 3 and represents the number of essential genes that can be controlled from that node.

Out of the seven top controlling proteins for the analysed breast cancer networks, four proteins are known to be of significance in breast cancer proliferation: *EGFR*²⁸, *AURKB*²⁹, *RAF1*³⁰ and *SRC*³¹. These proteins are targeted by several FDA-approved and investigational drugs used in fighting breast cancer (*lapatinib*, *vandetanib*, *canertinib*, *varlitinib*, *KX-01*, *dasatinib*). Additionally, two other drugs targeting the same proteins are investigated for use in treating unspecified cancer types (*LErafaON*, *XL281*). Two of the other top controlling proteins, *RET* and *MTCPI*, are known to be significant in other types of cancer, such as ovarian cancer³², pancreatic cancer³³, prostate cancer³⁴ or brain cancer³⁵. Even more, there already exists an FDA-approved breast cancer drug targeting these proteins (*selpercatinib*), as well as several approved and investigational drugs used for treating other cancer types (*ponatinib*, *sorafenib*, *sunitinib*, *cabozantinib*, *amuvatinib*). In particular, the drug *sorafenib*, used for the treatment of kidney and liver cancers, targets two of our top four controlling proteins, which may indicate its potential use in breast cancer as well. Indeed, there are several completed clinical trials researching the drug in the treatment of breast cancer³⁶. Additionally, the drug *cabozantinib* is also on trial for breast cancer treatment³⁷. Among the top controlling proteins was also *PLRG1*, for which no drug exists, marking it as a potential drug-target for future research.

For the analysed ovarian cancer networks, the algorithm identified six top controlling proteins. Three of these are documented as being of significance in ovarian cancer: *TGFB1*³⁸, *SRC*³⁹ and *MAPK3*⁴⁰, with several drugs targeting these proteins already approved or being under investigation for various types of cancer treatment (*dasatinib*, *XL228*, *seliciclib*, *ulixertinib*, *ponatinib*, *nintedanib*). Furthermore, the other three top controlling proteins, *SET*, *LCK* and *CSNK2A1*, are significant in other cancer types, such as leukemia⁴¹, colorectal cancer⁴² and prostate cancer⁴³.

For the pancreatic cancer networks analysed by the algorithm, we identified eleven top controlling proteins. Five of them are known to be significant in pancreatic cancer: *IGF1R*⁴⁴, *ERBB4*⁴⁵, *KRAS*⁴⁶, *ERBB2*⁴⁷ and *MTOR*⁴⁸ and targeted by several FDA-approved and investigational cancer drugs (*XL228*, *rhIGFB-3*, *linsitinib*, *brigatinib*, *SF1126*, *XL765*, *ridaforolimus*). Four of the other identified top controlling proteins, *ABL1*, *PDPK1*, *CSNK2A1* and *CDK1* are documented as significant in other types of cancer, for example breast cancer⁴⁹ and prostate cancer⁴³. They are, as well, targeted by several drugs under investigation for treating cancer (*AT-7519*, *seliciclib*). In particular, the drug *brigatinib* could be especially powerful in this case, as it targets four of the eleven top controlling proteins (*ABL1*, *IGF1R*, *ERBB4*, *ERBB2*). Another interesting case is the non-steroidal anti-inflammatory drug *celecoxib* targeting *PDPK1*, used to manage symptoms of arthritis pain and in familial adenomatous polyposis, which has also been under investigation as potential cancer chemo-preventive and therapeutic drug⁵⁰. Indeed, there are several ongoing and completed trials researching the effect of the drug in pancreatic cancer⁵¹. Among the top controlling proteins were also *PPP2R1A* and *DUSP7*, for which no drug exists, marking them potential drug-targets for future research.

Furthermore, we found the drug *fostamatinib*, used for the treatment of rheumatoid arthritis and immune thrombocytopenic purpura, which targets five out of the seven top controlling proteins for breast cancer, four out of the six for ovarian cancer, and seven out of the eleven for pancreatic cancer. Our algorithm thus suggests that the drug could potentially be used in cancer treatment. This idea is supported by several completed clinical trials for using *fostamatinib* in treating lymphoma⁵² and one ongoing trial for ovarian cancer⁵³.

Discussion

Applying network controllability for drug repurposing is not yet well-established. It is a very promising line of research that is currently hindered in part by the lack of powerful implementations of network controllability algorithms. This is where this paper and its algorithm contribute, making it possible to run detailed drug repurposing studies based on network controllability. We applied the algorithm for a few relatively small cancer examples, demonstrating its feasibility in the medical domain. Our demonstration was intended to show the potential of the network controllability approach to help in such studies. Our case studies suggest that this approach is promising in drug repurposing: we identified several approved drugs whose targets

contribute to controlling the essential genes specific to other diseases than those the drugs were approved for. These results are to be considered as proof-of-concept, rather than fully fledged validated demonstrations of network control-based drug repurposing identification. Our search strategy was based on a genetic algorithm, where the population in each generation of the training of the algorithm is a set of valid solutions to the network controllability problem. The algorithm turned out to be scalable, with its performance staying strong even for very large networks.

The genetic algorithm comes by design with a set-limit on the maximum length of control paths from the input to the target nodes. We made this into a parameter whose value can be set by the user and we tested it with a value of 5. Having this free parameter is a feature of particularly important interest in applications in medicine, where the effects of a drug dissipate quickly over longer signalling paths. The focused search upstream of the target nodes led to the genetic algorithm drastically improving the percentage of FDA-approved drug targets selected in its solution, a clear step forward towards applications in combinatorial drug selection and drug repurposing. The drugs identified by our algorithm as potentially efficient for breast, ovarian, and pancreatic cancer correlate well with recent literature results, and some of our suggestions have already been subject to several clinical studies. This strengthens the potential of our approach for studies in synthetic lethality-driven drug repurposing.

Methods

Network controllability

We introduce briefly in the supplementary information the basic concepts of target structural network controllability and the Kalman condition for the problem.

The outline of the genetic algorithm

An overview of the basic outline of the algorithm is presented in figure 1. The algorithm takes as input a network given as a directed graph $G = (V, E)$ and a list of target nodes $T \subseteq V$, $T = \{t_1, \dots, t_l\}$. We denote the graph's adjacency matrix by A_G . The algorithm gives as a result a set of input nodes $I \subseteq V$ controlling the set T , with the objective being to minimize the size of I . The algorithm can also take as an additional, optional input a set $P \subseteq V$ of so-called preferred nodes. In this case, the algorithm will aim for a double optimization objective: minimize the set I , while maximizing the number of elements from P included in I . Our typical application scenario is that of a network G consisting of directed protein-protein interactions specific to a disease mechanism of interest, with the set of targets T being a disease-specific set of essential genes, and the set of preferred nodes P a set of proteins targetable by available drugs or by specially designed compounds (e.g., inhibitors, small silencing molecules, etc.) The terminology we use to describe the algorithm, e.g., population/chromosome/crossover/mutation/fitness is standard in the genetic algorithm literature and refers to its conventions, rather than being suggestive of specifics in molecular biology.

Our algorithm starts by generating several solutions to the control problem, in the form of several *control sets* $I_1, \dots, I_m$; we discuss how this is achieved in the supplementary information. Each such solution is encoded as a “chromosome”, i.e., a vector of “genes” $[g_1, \dots, g_l]$, where for all $1 \leq i \leq l$, $g_i \in V$ controls the target node $t_i \in T$. In particular, g_i is an ancestor of t_i in graph G , for all $1 \leq i \leq l$. Note that a node can control simultaneously several other nodes in the network and so, the genes $g_1, \dots, g_l$ of a chromosome are not necessarily distinct. In fact, the fewer the genes on a chromosome, the better its fitness will be, as we discuss in the supplementary information. The algorithm also implements a preference in the choice of preferred nodes as genes on the chromosomes; the details are discussed in the supplementary information.

A set of chromosomes is called a population. Note that a chromosome will always encode a solution to our optimization problem, throughout the iterative run of the algorithm. Any population maintained by the algorithm consists of several such chromosomes, some better than others from the point of view of our optimization criteria, but all valid solutions to the target controllability problem to be solved.

The algorithm iteratively generates successive populations (sets of chromosomes) that get better at the optimization it aims to solve: the number of distinct genes on some chromosomes gets smaller and the proportion of preferred nodes among them gets higher. The algorithm stops after a number of iterations in which the quality of the solution does not get improved. To have a bounded running time even for large networks, we also added a stop condition after a maximum number of iterations; this was never reached during our tests. This pre-defined stop is necessary since the target structural controllability problem is known to be NP-hard and so, finding the optimal solution can require a prohibitively high number of steps, potentially exponential in the number of nodes in the network. The output consists of several solutions to the problem, represented by all the control sets in the final population obtained by the algorithm.

The initial population of solutions is randomly generated in such a way that each element selected for it is a solution to the target structural controllability problem (A_G, I, T) . To generate the next generation from the current one, we use three techniques:

- Retain in the population the best solutions (from the point of view of the assessed optimization problem, as encoded in the fitness function). “Elitism” is used to conserve the best solutions (further discussed in the supplementary information).

- Add random chromosomes (all being valid solutions to the optimization problem, albeit potentially of lower fitness score than some of the others in the population).
- Generate new solutions/chromosomes resulting from combinations of those in the current population. A selection operator is used to choose the chromosomes which will produce offsprings for the following generation. New chromosomes are produced using crossover and mutation (discussed in details in the supplementary information).

A list of all the parameters used by the genetic algorithm can be found in table 1. The basic outline of the proposed genetic algorithm is described below. All operators are detailed in the supplementary information.

1. *Generate the initial population.* We set $t \leftarrow 0$ for the first generation. We initialize $P(t)$ with a number of n randomly generated chromosomes.
2. *Preserve the fittest chromosomes.* We evaluate the fitness of all chromosomes in $P(t)$. We add to the next population P_{t+1} the $p_e \cdot n$ chromosomes in the current generation with the highest fitness score, where $0 \leq p_e < 1$ is the ‘elitism’ parameter. If there are more chromosomes of equal fitness, the ones to be added are randomly chosen.
3. *Add random chromosomes.* We add $p_r \cdot n$ new randomly generated chromosomes to $P(t+1)$, where $0 \leq p_r < 1$ is the ‘randomness’ parameter.
4. *Add the offsprings of the current population.* We apply two times the selection operator on $P(t)$, obtaining two chromosomes of P_t selected randomly with a probability proportional to their fitness score. On the two selected chromosomes we apply the crossover operator, obtaining an offspring to be added to $P(t+1)$. The offspring is added in a mutated form with the mutation probability $0 \leq p_m < 1$. We continue applying this step until the number of chromosomes in $P(t+1)$ becomes n .
5. *Iterate.* If the current index $t < N$, the maximum number of generations, we set $t \leftarrow t + 1$ and continue with Step 2.
6. *Output.* We return the fittest chromosomes in the current generation as solutions to the problem and stop.

References

1. Kitano, H. Computational systems biology. *Nature* **420**, 206–210 (2002).
2. Tian, Q., Price, N. D. & Hood, L. Systems cancer medicine: towards realization of predictive, preventive, personalized and participatory (p4) medicine. *J Intern Med* **271**, 111–121 (2012).
3. Barabási, A.-L., Gulbahce, N. & Loscalzo, J. Network medicine: a network-based approach to human disease. *Nat. reviews. Genet.* **12**, 56–68 (2011).
4. Goh, K.-I. *et al.* The human disease network. *Proc. Natl. Acad. Sci.* **104**, 8685 (2007).
5. Czeizler, E., Gratie, C., Kai Chiu, W., Kanhaiya, K. & Petre, I. Structural target controllability of linear networks. *IEEE/ACM Transactions on Comput. Biol. Bioinforma.* **15**, 1217–1228 (2018).
6. Kanhaiya, K., Czeizler, E., Gratie, C. & Petre, I. Controlling directed protein interaction networks in cancer. *Sci. Reports* **7** (2017).
7. Ochsner, S. A. *et al.* The signaling pathways project, an integrated ‘omics knowledgebase for mammalian cellular signaling pathways. *Sci. Data* **6**, 252 (2019).
8. Misselbeck, K. *et al.* A network-based approach to identify deregulated pathways and drug effects in metabolic syndrome. *Nat. Commun.* **10**, 5215 (2019).
9. Davis, M. M., Tato, C. M. & Furman, D. Systems immunology: just getting started. *Nat. immunology* **18**, 725–732 (2017).
10. Liu, X. *et al.* Computational methods for identifying the critical nodes in biological networks. *Briefings Bioinforma.* **21**, 486–497 (2019).
11. Cheng, F., Kovács, I. A. & Barabási, A.-L. Network-based prediction of drug combinations. *Nat. Commun.* **10**, 1197 (2019).
12. Kalman, R. E., Ho, Y. C. & Narendra, K. S. Controllability of linear dynamical systems. *Contributions to Differ. Equations* **1**, 189–213 (1963).
13. Liu, Y.-Y., Slotine, J.-J. & Barabási, A.-L. Controllability of complex networks. *Nature* (2011).

14. Gao, J., Liu, Y.-Y., D'Souza, R. M. & Barabási, A.-L. Target control of complex networks. *Nat. Commun.* **5**, 5415 (2014).
15. Molnár, F., Sreenivasan, S., Szymanski, B. K. & Korniss, G. Minimum dominating sets in scale-free network ensembles. *Sci. Reports* **3**, 1736 (2013).
16. Rancati, G., Moffat, J., Typas, A. & Pavelka, N. Emerging and evolving concepts in gene essentiality. *Nat. Rev. Genet.* **19**, 34–49 (2018).
17. Whitley, D. & Sutton, A. M. Genetic algorithms — a survey of models and methods. In Rozenberg, G., Bäck, T. & Kok, J. N. (eds.) *Handbook of Natural Computing*, 637–671 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).
18. Koh, J. *et al.* Colt-cancer: functional genetic screening resource for essential genes in human cancer cell lines. *Nucleic acids research* **40**, D957–D963 (2011).
19. Milo, R. *et al.* Network motifs: Simple building blocks of complex networks. *Science* **298**, 824–827 (2002).
20. Milo, R. *et al.* Superfamilies of evolved and designed networks. *Science* **303**, 1538–1542 (2004).
21. Varoquaux, G., Vaught, T. & Millman, J. (eds.). *Exploring network structure, dynamics, and function using NetworkX*, Proceedings of the 7th Python in Science Conference (SciPy2008) (2008).
22. Wishart, D. S. *et al.* Drugbank 5.0: A major update to the drugbank database for 2018. *Nucleic Acids Res.* **46**, D1074–D1082 (2018).
23. Popescu, V. GeneticAlgNetControl (2020). Available at <https://github.com/vicbgdn/GeneticAlgNetControl>, version 1.0.
24. The UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Res.* **43**, D204–D212, DOI: [10.1093/nar/gku989](https://doi.org/10.1093/nar/gku989) (2014).
25. Perfetto, L. *et al.* SIGNOR: a database of causal relationships between biological entities. *Nucleic Acids Res.* **44**, D548–D554, DOI: [10.1093/nar/gkv1048](https://doi.org/10.1093/nar/gkv1048) (2015).
26. Moradi, M., Goliaei, S. & Foroughmand-Araabi, M.-H. A boolean network control algorithm guided by forward dynamic programming. *PLOS ONE* **14**, e0215449– (2019).
27. Baudin, A., Paul, S., Su, C. & Pang, J. Controlling large boolean networks with single-step perturbations. *Bioinformatics* **35**, i558–i567, DOI: [10.1093/bioinformatics/btz371](https://doi.org/10.1093/bioinformatics/btz371) (2019).
28. Burness, M. L., Grushko, T. A. & Olopade, O. I. Epidermal growth factor receptor in triple-negative and basal-like breast cancer: Promising clinical target or only a marker? *Cancer journal* **16**, 23–32 (2010).
29. Gully, C. P. *et al.* Antineoplastic effects of an aurora b kinase inhibitor in breast cancer. *Mol. cancer* **22**, 9–42 (2010).
30. Li, H. Z. *et al.* Effects of raf kinase inhibitor protein expression on metastasis and progression of human breast cancer. *Mol. cancer research* **7**, 832–840 (2009).
31. Morgan, L., Nicholson, R. I. & Hiscox, S. Src as a therapeutic target in breast cancer. *Endocrine, metabolic immune disorders drug targets* **8**, 273–278 (2008).
32. Guan, L. *et al.* Oncogenic and drug-sensitive ret mutations in human epithelial ovarian cancer. *J. experimental clinical cancer research* **39** (2020).
33. Donahue, T. R. & Hines, O. J. Cxcr2 and ret single nucleotide polymorphisms in pancreatic cancer. *World journal surgery* **33**, 710–715 (2009).
34. VanDeusen, H. R. *et al.* Targeting ret kinase in neuroendocrine prostate cancer. *Mol. Cancer Res.* (2020).
35. Han, L., Liu, H., Wu, J. & Liu, J. mir-126 suppresses invasion and migration of malignant glioma by targeting mature t cell proliferation 1 (mtcp1). *Med. science monitor* **24**, 6630–6637 (2018).
36. City of Hope Medical Center. Sorafenib and vinorelbine in treating women with stage iv breast cancer (2014). Available at <https://clinicaltrials.gov/ct2/show/NCT00828074>.
37. Dana-Farber Cancer Institute. Cabozantinib for metastatic triple negative brca (2016). Available at <https://clinicaltrials.gov/ct2/show/NCT01738438>.
38. Sharma, A. *et al.* Effects of omega-3 fatty acids on components of the transforming growth factor beta-1 pathway: Implication for dietary modification and prevention in ovarian cancer. *Am. journal obstetrics gynecology* **200**, 516.e1–516.36 (2009).
39. Han, L. Y. *et al.* Antiangiogenic and antitumor effects of src inhibition in ovarian carcinoma. *Cancer research* **66**, 8633–8639 (2006).

40. Yu, T.-T., Wang, C.-Y. & Tong, R. Erbb2 gene expression silencing involved in ovarian cancer cell migration and invasion through mediating mapk1/mapk3 signaling pathway. *Eur. review for medical pharmacological sciences* **24**, 5267–5280 (2020).
41. Jiang, G. *et al.* In human leukemia cells ephrin-b-induced invasive activity is supported by lck and is associated with reassembling of lipid raft signaling complexes. *Mol. cancer research* **6**, 291–305 (2008).
42. Cristóbal, I. *et al.* Deregulation of set is associated with tumor progression and predicts adverse outcome in patients with early-stage colorectal cancer. *J. clinical medicine* **8** (2019).
43. Ahmad, K. A., Wang, G. & Ahmed, K. Intracellular hydrogen peroxide production is an upstream event in apoptosis induced by down-regulation of casein kinase 2 in prostate cancer cells. *Mol. cancer research* **4**, 331–338 (2006).
44. Zheng, Y. *et al.* Insulin-like growth factor 1-induced enolase 2 deacetylation by hdac3 promotes metastasis of pancreatic cancer. *Signal transduction targeted therapy* **5**, 53 (2020).
45. Hedegger, K. *et al.* Unraveling erbb network dynamics upon betacellulin signaling in pancreatic ductal adenocarcinoma in mice. *Mol. oncology* (2020).
46. Shankar, S. *et al.* An essential role for argonaute 2 in egfr-kras signaling in pancreatic cancer development. *Nat. communications* **11**, 2817 (2020).
47. Booth, L., Poklepovic, A. & Dent, P. Neratinib decreases pro-survival responses of [sorafenib + vorinostat] in pancreatic cancer. *Biochem. pharmacology* **178**, 114067 (2020).
48. Zhao, H.-W., Zhou, N., Jin, F., Wang, R. & Zhao, J.-Q. Metformin reduces pancreatic cancer cell proliferation and increases apoptosis through mtor signaling pathway and its dose-effect relationship. *Eur. review for medical pharmacological sciences* **24**, 5336–5344 (2020).
49. Johnson, N. *et al.* Pre-clinical evaluation of cyclin-dependent kinase 2 and 1 inhibition in anti-estrogen-sensitive and resistant breast cancer cells. *Br. journal cancer* **102**, 342–350 (2010).
50. Gong, L. *et al.* Celecoxib pathways: Pharmacokinetics and pharmacodynamics. *Pharmacogenetics genomics* **22**, 310–318 (2012).
51. Second Affiliated Hospital, School of Medicine, Zhejiang University. Gemcitabine and celecoxib combination therapy in treating patients with r0 resection pancreatic cancer (gcrp) (2023). Available at <https://clinicaltrials.gov/ct2/show/NCT03498326>.
52. AstraZeneca. Study to learn if 200mg test drug (fostamatinib) helps people with large b-cell lymphoma, a type of blood cancer (2013). Available at <https://clinicaltrials.gov/ct2/show/NCT01499303>.
53. Sidney Kimmel Comprehensive Cancer Center at Johns Hopkins. Clinical trial of combined fostamatinib and paclitaxel in ovarian cancer (2020). Available at <https://clinicaltrials.gov/ct2/show/NCT03246074>.

Acknowledgments

This work was partially supported by the Academy of Finland (project 311371/2017 to E.C.) and by the Romanian National Authority for Scientific Research and Innovation (PED grant 2391 to I.P.).

Author contributions statement

V-B.P., I.N., E.C. and I.P. designed the study. V-B.P. and I.P. performed the computational analysis. V-B.P. and K.K. collected and curated the data. V-B.P., E.C. and I.P. analyzed the results and co-wrote the manuscript. All authors were involved in drafting the article and all have reviewed and approved the final version of the manuscript.

Additional information

Competing interests. The author(s) declare no competing interests.

Data availability. All of the data and algorithm results are available in the application repository at²³.

Figure 1. The basic outline of the genetic algorithm, shown in a clockwise manner: the required data, the initial setup and chromosome encoding, the operators of the genetic algorithm, and the chromosome decoding and final result.

Table 1. The parameters used by the genetic algorithm

Parameter	Meaning	Default
$N \in \mathbb{N}_{\geq 1}$	Total number of generations for which the algorithm runs	$N = 10000$
$n \in \mathbb{N}_{\geq 2}$	Total number of chromosomes in a generation	$n = 80$
$p_m \in \mathbb{R}_{[0,1]}$	Probability of mutation for a chromosome	$p_m = 0.01$
$p_e \in \mathbb{R}_{[0,1]}$	Maximum percentage of elites in a generation	$p_e = 0.25$
$p_r \in \mathbb{R}_{[0,1]}$	Percentage of randomly generated chromosomes in a generation	$p_r = 0.25$
$\max_{path} \in \mathbb{N}_{\geq 1}$	Max. number of interactions in a control path	$\max_{path} = 5$
$\max_{rand} \in \mathbb{N}_{\geq 1}$	Max. number of randomly generated genes in a chromosome	$\max_{rand} = 15$

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SISGenAlg4NetControl.pdf](#)