Tel-Aviv University

Raymond and Beverly Sackler Faculty of Exact Sciences

The Blavatnik School of Computer Science

Personalized prioritization of driver genes

Thesis submitted in partial fulfillment of graduate requirements for

The degree "Master of Sciences" in Tel-Aviv University

School of Computer Science

By

Gal Dinstag

Prepared under the supervision of

Prof. Ron Shamir

December 2018

Acknowledgements

I would to express my deep gratitude to the people I have worked with in this journey and those who made it happen directly and indirectly.

First and foremost to my supervisor Prof. Ron Shamir, a great scientist and a distinguished teacher who taught me that good science is a constant effort for improvement that should be done carefully and with a critical eye. Despite the long research subject exploration that sometimes seemed like heading towards a dead end, Ron gave me the opportunity to peruse the goals I was interested in and I am grateful to him for that. I would also like to thank him for helping me spread our science in domestic venues and abroad even when I was doubtful and for all his financial support from the very beginning to the very end. I am proud of the work we have done and honored to be his student.

Second, I would like to thank Dr. David Amar for collaborating with me in my first project in the lab and for sharing his amazing scientific skills with me. His sharp mind, thorough methodology and persistent drive for excellence helped to shape the way I (am trying to) do science. I am grateful for the opportunity to work with him and positive that great things are yet to come for him.

Third, I would like to thank my lab: Tom, Dvir, David, Nimrod, Ron Z., Idan, Dan, Hagai, Neta, Liad, Aviv, Yael and Roi for being my mates in this amazing way, for their sympathetic ear for science and other "life itself" manners, for our fruitful discussions and above all, for making this ride joyful. Special thanks to Gilit Zohar-Oren for all her super fast support in every issue big or small along the way. I wouldn't have made it without you all.

I would like to deeply thank the agencies that supported my thesis research: The Safra Center for Bioinformatics at Tel Aviv University, the Bella Walter Memorial Fund of the Israel Cancer Association, the Israel Science Foundation as part of the ISF-NSFC joint program (grant 2193/15) and Len Blavatnik and the Blavatnik Family foundation.

Last but not least to my beloved ones: my parents Yael and Moti who seeded my curiosity and eager for knowledge and demonstrated the importance of education and hard work, and my fiancé Nofar for being my partner to life, supporting me the whole way and putting up with my digging.

This thesis is dedicated to the memory of my father,

Moti Dinstag (1963-2018)

You are always in my heart

Table of Contents

Abstract

Evolution of cancer is driven by few somatic mutations that disrupt cellular processes, causing abnormal proliferation and tumor development, while most somatic mutations have no impact on progression. Distinguishing those mutated genes that drive tumorigenesis in a patient is a primary goal in cancer therapy: Knowledge of these genes and the pathways on which they operate can illuminate disease mechanisms and indicate potential therapies and drug targets. Current research focuses mainly on cohort-level driver gene identification, but patient-specific driver gene identification remains a challenge.

We developed a new algorithm for patient-specific ranking of driver genes. The algorithm, called PRODIGY, analyzes the expression and mutation profiles of the patient along with data on known pathways and protein-protein interactions. Prodigy quantifies the impact of each mutated gene on every deregulated pathway using the prize collecting Steiner tree model. Mutated genes are ranked by their aggregated impact on all deregulated pathways.

In testing on five TCGA cancer cohorts spanning >2500 patients and comparison to validated driver genes, Prodigy outperformed extant methods and did better than rankings based on network centrality measures. Our results emphasize the pleiotropic effect of driver genes and show that Prodigy is capable of identifying even very rare drivers. Hence, Prodigy can assist oncologists in decisions regarding personalized treatment.

1. Biological background

1.1 Cancer

Cancer is a complex disease that encompasses more than 100 distinct diseases with diverse risk factors and prognosis. It originates in many cell types and organs in the body. It is caused by genomic and epigenomic alterations that accumulate in a normal cell, turning it into a cancer cell. Cancer cells are characterized by extensive proliferation that leads to the formation of tumors that penetrate normal tissues and form metastases in distant organs[1].

1.1.1 Mutational landscape of cancer

Most cancers are mutation driven. The genomic alterations that cause the cancerous processes include, among others, single nucleotide variations and small DNA insertions/deletions (termed *SNV*), translocations (exchange of two end fragments of chromosomes), segment inversions and copy number variations (CNV) due to deletion or amplification of DNA segments, and even whole chromosomes. Epigenomic alterations are changes in the DNA molecules and in proteins that interact with the DNA that are not manifested in the DNA sequence itself. They can cause changes in the DNA structure in the cell through rearrangements in the DNA packing (e.g. by changes in the chromatin structure) as a result of histone modifications, or changes in the chemical formation of nucleotides using methylation (the addition of a methyl group to cytosine nucleotides) that can alter transcription. Although it was shown that epigenetic events can drive carciogenesis[2], these events are far less explored than DNA mutations and we will not discuss them further in this work.

About 90% of the genomic alterations in known cancer genes occur in somatic cells, while ~20% occur in germline cells and ~10% occur in both[3]. The overall number of observed mutations varies among tumor tissues. Kim and Kim[4] analyzed dozens of cancer patient cohorts from TCGA[5] and found that the average number of somatic mutations can reach up to thousands per tumor in some cancer subtypes. They also showed huge variation among cancers in the number of mutations, and this finding was corroborated in other studies[6]. In addition, variation in the number of mutations was also shown to arise from environmental factors like smoking[7], exposure to UV light[6] and age[8].

1.1.2 Cancer is an evolutionary process

Cancer is the outcome of a Darwinian evolution process occurring in cell populations. Analogous to Darwinian evolution occurring in species, cancer development is based on the accumulation of mutations over time, granting the cancer cell two crucial features: extensive proliferation ability and a selective advantage over its microenvironment. Cells that harbor destructive mutations are eliminated through cell death, and cells that carry mutations beneficial to cell survival are positively selected in the microenvironment. There are also mutations neutral to the cell functioning acquired during tumorigenesis and we will discuss them later. Somatic mutations are not exclusive to cancer and they also happen in healthy cells, as the acquisition of mutations is more or less random. If those mutations confer only limited abnormal growth advantage, the cells may form benign tumors that are pathologically invisible or manifest as common benign growths, e.g., skin moles. In the more severe case, the advantageous cell acquires a set of mutations that allow it to go through extreme proliferation and develop into malignant tumors[1].

Vogelstein & Kinzler[9] describe cancer as a "three strikes" process that is directed by three mutational events (**Figure 1**): (1) the "breakthrough phase", in which a cell acquires a mutation and begins to proliferate abnormally. It takes many years for the cells resulting from this proliferation to be observable clinically, if they ever are. (2) The "expansion phase" in which a second mutation enables the cell to thrive in its local environment despite low concentrations of growth factors, nutrients, oxygen, and appropriate cell-to-cell contacts. (3) A third mutation enables cells to invade normal tissues and grow in an otherwise hostile environments. We will now discuss the nature of those mutations and their uniqueness compared to other mutational events.

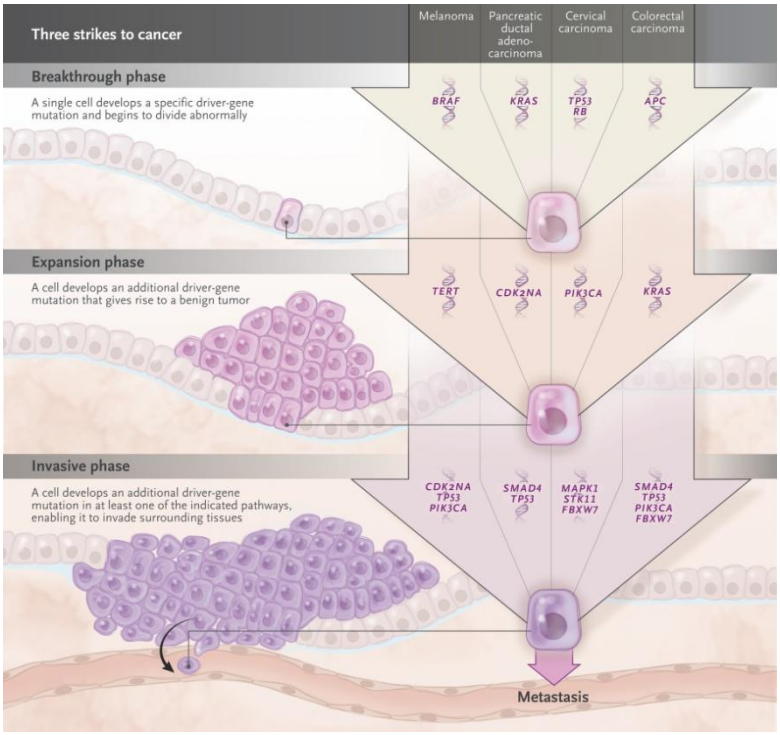


Figure 1: The "three strikes" of cancer: The first mutation provides initial growth advantage ("Breakthrough phase"), a second mutation accelerates proliferation ("Expansion phase"), and a third mutation gives rise to metastasis ("Invasive phase"). Source: Vogelstein & Kinzler[9]

1.1.3 Driver genes

As discussed above, somatic mutations are not unique to cancer cells; normal cells also acquire random somatic mutations. Some of them are neutral (i.e. do not alter cell mechanisms at all) and so do not require repair, while others impair natural functions in the cell, forcing it to go through DNA repair or cell death if the repair fails[6,10,11]. In both cases, the normal cell is not transformed into a cancer cell.

The difference between the evolving cancerous cell and the normal cell are in the post-somatic mutation acquisition phase: in the former, the cell gains new growth or selective advantage it did not possess before, and in the latter it is either impaired or neutral to the mutation and does not gain any growth advantage.

Driver mutations: Mutational events that grant such advantages to the cell and "drive" it into tumorigenesis are called *driver mutations* (or driver events) and the genes in which these mutations take place are called *driver genes*. In contrast, *passenger mutations* are acquired extensively during cancer progression simply because cancer cells over-proliferate in orders

of magnitude more compared to normal cells, and random mutations mainly occur during cell division. These mutations do not provide any of the advantages of driver mutations.

There is a very extensive debate regarding the number of driver mutations among the observed mutations in each tumor[1,6,12], but the consensus is that this number is very low. Obviously, there are many factors that contribute to the variation in the number of drivers, including the progression stage of the tumor[9], its tissue of origin[13], environmental properties such as smoking[7] and other factors like age[14]. Tomasetti et al.[15] showed that as little as three driver mutations suffice to develop lung and colorectal cancer. Nordling[16] and Armitage[17] suggested six or seven as the typical number of drivers.

1.1.4 Distinguishing drivers from passengers

It is therefore a challenge to distinguish driver from passenger mutations. The need to do so has high priority in cancer research - and in personalized cancer medicine in particular - for several reasons: 1) knowledge of the drivers and the mechanisms by which they operate can suggest potential treatments and drug targets. 2) Basing cancer treatment on molecular signatures rather than on the disease organ offers the opportunity to treat individuals with regimens not yet considered for their specific type of cancer. For example, many "basket" clinical trials, in which a specific drug is given to patients with diverse cancer types based on specific biomarkers, show that the same drug can sometimes have high efficiency across different cancers if the right mutation is detected[18] .

1.2 Protein-protein interactions

Protein-protein interactions (PPIs) allow proteins to stably or transiently work together. When stably linked, PPIs form the basis of the quaternary structure of proteins (i.e. protein complexes made of several protein chains). PPIs also describe very short temporal connections between proteins for functional reasons like phosphorylation or activation. Svedberg[19,20] first established that some proteins form complex organizations. Further research developed PPI detection techniques, and experiments that utilized them led to the discovery of large numbers of PPIs[21]. The key point is that the three-dimensional structure of some proteins may become meaningful only in the context of a larger protein

assembly. Such protein complexes are found in every cellular location, in organelles, the cytosol and the cell membranes[22]. They are of great biological importance as they mediate most biochemical reactions in the cell, including enzyme-substrate functions, signal transduction and protein degradation.

PPIs are sometimes defined more broadly to include an interaction of a protein to a non-protein. For example, a protein-DNA interaction that describes a binding between a transcription factor *A* and its target binding site in the promoter of gene *B*, can be presented as a PPI between *A* and *B.* Another example is the interaction between a protein and chemical molecules that serve as cofactors or substrates. Some PPIs are inferred from computational studies that hypothesize an interaction between two proteins based on genomic data. For example, one might predict an interaction between two genes that are in close proximity in the genome and are co-conserved throughout evolution even without experimental evidence. Another common practice is to consider a functional interaction between genes or proteins (e.g. if the two participate in the same pathway) as PPI even if the two proteins do not physically interact. The interaction between two entities *X* and *Y* can be directed (e.g., *X* is applied on *Y* as activator/inhibitor/modifier etc.) or undirected (e.g., *X* and *Y* bind to form a complex).

PPIs can be identified experimentally or predicted using computational methods. Experimental techniques are divided into large scale methods that can identify many interactions at once, and focused experiments that examine a few or a single interaction at a time.

1.2.1 Experimental methods for PPI identification

The two-hybrid system[23] is a high-throughput method that uses transcriptional activity to reveal PPIs. It exploits the natural mechanism of many transcriptional activation modules, which consists of a DNA binding domain and a transcriptional activation domain. These two domains are needed to be in contact or close proximity for the transcription of the target gene to occur. Contact can be formed using a single protein that contains both appropriate domains or by two proteins, each containing one of the domains. For simplicity, we will describe the case where there are two different activating proteins. The first activator protein binds the DNA-binding domain, and the second activator protein attaches to the first and recruits other proteins needed for transcription through its transcription activation site. The two-hybrid

system leverages the fact that the two activator proteins can be brought to proximity by the interaction of any two proteins.

This system requires two hybrids to be constructed (**Figure 2)**: a DNA-binding domain fused to some protein *X*, and a transcription activation domain fused to some protein *Y*. *X* and *Y* are the putative interacting proteins we wish to explore. These two hybrids are expressed in a cell containing one or more reporter genes that are naturally activated by the specific machinery (i.e. the specific DNA-binding and transcription domains). If *X* and *Y* interact, the activation domain is brought into close proximity with the DNA-binding domain, leading to the expression of the reporter gene, which can be detected using a sensor (fluorescence for example). This is a very well established method and one of the first large scale techniques for PPI identification. However, it was shown that this method produces high frequency of false positive interactions[24].

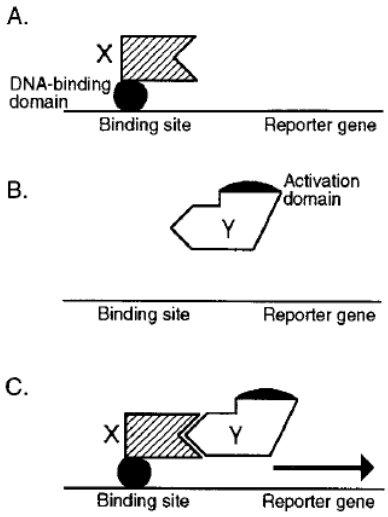


Figure 2: The two-hybrid system. The two proteins in question *X* and *Y* are fused to two different proteins, each containing either DNA-binding domain or a transcription activation domain of one or more reporter genes. (A) The DNA-binding domain hybrid does not activate transcription because *X* does not contain an activation domain. (B) The activation domain hybrid does not activate transcription because *Y* does not attach to the DNA-binding site. (C) Interaction between *X* and *Y* brings the activation domain into close proximity with the DNA-binding site and results in transcription of the reporter gene. Sorece: Phizicky and Fields[25]

Another high throughput method for PPI detection is the protein microarray[26]. This method is used analogously to DNA and mRNA microarrays. In this approach, target proteins are first fluorescently labeled and then covalently attached to chemically derivatized glass slides at extremely high density. The proteins are fixed in a way that preserves their folded

conformations, allowing them to interact with other proteins in their natural form. Candidate interacting partners of the target proteins are then applied on the array, and successful interactions are identified using a florescence scanner.

1.2.2 Computational methods for PPI identification

In addition to experimental methods, many algorithms were developed in order to predict PPIs and their predictions populate many PPI databases[27].

Phylogenetic profiling is a computational method to predict PPIs based on the detection of gene pairs that share a similar species profile. That is, they are present or absent together in the same species[28]. The idea behind this approach is that proteins that need each other to perform a given function will either be simultaneously present or absent in a specie. The two proteins need not vanish evolutionarily at the same time and can be a result of "reductive evolution": an organism (especially bacteria) might remove a certain gene if their corresponding partner was previously revoked. The opposite case where the two genes are gained sequentially is far less probable. As a result, interacting or functionally related genes would tend to have similar presence profiles.

Another phylogeny-driven approach for PPI prediction focuses on the coevolution of a pair of genes rather than their co-presence or absence. Pazos and Valencia[29] introduced a method that predicts PPIs based on the similarity between the phylogenetic trees of the appropriate genes. The hypothesis is that interacting proteins would be subjected to coevolution, which would be manifested in highly similar phylogenetic trees. The similarity between the phylogenetic trees is measured using a distance metric. By comparison to a null distribution of phylogenetic distances from random gene pairs, those that exhibit a larger similarity than expected by chance are identified as interacting pairs.

Gene fusion is a method that infers interaction between genes that are fused in some genomes and are not fused in others[30]. This method stems from the observation that some interacting proteins are encoded by the same (fused) gene in one organism and by different genes in others. In some cases, it will be logical to infer that the two proteins are physically or functionally related. However, these events are not very frequent in eukaryotes.

Mining the biological literature is also a common practice in PPI detection. These methods use natural language processing to retrieve putative interactions from biological publications: Thomas et al.[31] parsed abstracts of biological publications to detect interacting proteins. Andrea et al.[32] conducted frequency analysis of individual words in abstracts to come up with interacting proteins. Marcotte et al.[33] developed a method that first predicts whether a given paper addresses PPIs using the frequencies of discriminating words found in the abstract and then mines the interactions from the paper using the two previous methods.

Lastly, co-expression of genes is also often used when predicting PPIs. It exploits the fact that genes whose proteins participate in the same pathway or are part of the same protein complex are often co-regulated under a large number of diverse conditions. On the other hand, it is possible for co-expressed genes to have similar regulation without being functionally related. In order to circumvent this confounding effect, Stuart et al.[34] used also evolutionary conservation information to derive novel interactions.

1.2.3 Main PPI databases

1.2.3.1 STRING

STRING[35] (Search Tool for the Retrieval of Interacting Genes/Proteins) aims to collect, predict and unify most types of PPIs, including direct and indirect associations as discussed above. Each interaction in the database is annotated with a numerical confidence score, which can be used to filter them. The tradeoff is between retaining highly reliable interactions (by setting a higher confidence threshold) and including more interactions. The interaction information is freely available for download. STRING interactions come from the consolidation of other PPI sources, physical interactions from experimental datasets, and from predicted interactions based on four methods: phylogenetic analysis, gene fusions, text mining and co-expression. STRING is the largest database for human interactions and it contains >4.5 million scored interactions in human alone.

1.2.3.2 ReactomeFI

A PPI network constructed by Wu et al.[36] in 2010 was adopted by many recent computational methods, including two methods that will be discussed later in chapter 3. It constructed a network of functional interactions (FIs) in order to help studies that identify candidate disease genes. The authors introduced a naïve Bayes classifier to distinguish high-likelihood FIs from

non-functional pairwise relationships and false positive interactions. The classifier was trained on curated interactions from Reactome pathways (described below) and predicted FIs from physical sources of PPIs in human and model organisms, gene co-expression data, protein domain-domain interactions, protein interactions generated from text mining and GO annotations. Overall, the ReactomeFI network contains 11,648 nodes and 211,794 directed unweighted edges.

1.3 Cellular Pathways

Cellular pathways describe the processes and mechanisms by which the cell operates. Examples include the cell cycle and cell death, metabolic processes, protein degradation, signal transduction etc. They are mostly composed of proteins. Simple pathways are built as cascades: one or more "entry points" (the first protein in the cascade) are activated and subsequently activate the next protein in the chain and so on, until a final product is produced. The final product of the pathway can take many forms such as small molecules or metabolites, activation of a target protein or expression of a gene. However, pathways are usually not acyclic. They contain internal feedback loops and revertible checkpoints, require the formation of complexes and include cycles. Also, they may contain different types of "players" other than simple proteins, e.g., complexes and chemical compounds. Pathways do not work in isolation but cross-talk with each other, constructing higher-order cascades and feedback loops, so that separation of pathways is somewhat artificial.

Pathways can be visualized as graphs, where each entity is represented by a node and edges describe the relations between the different entities (e.g. binding of complexes, activation or inhibition). An example of the MAPK pathway (a process that promotes cell proliferation and is important in many cancers) in graph form is shown in **Figure 3**.

1.3.1 Pathways and complex diseases

Normal cellular division is tightly controlled by a complex network of signaling pathways that ensures that cells proliferate only when required to by the body as a whole (e.g. during development or wound healing). Cancer occurs when these regulations break down, among other reasons because of defects in these signaling pathways. An example of the perturbation of these pathways is through mutations in *RAS* proteins[38]. RAS proteins have essential roles in the activity of several crucial signaling pathways that regulate cellular

proliferation (including the MAPK pathway shown in Figure 3). Moreover, RAS proteins that were perturbed with a point mutation are common in tumors. These RAS variants are constantly active, resulting in constant activation of proliferation, accelerating tumor development. RAS activity also promotes invasiveness of the tumor cells and the ability to induce new blood vessel formation[39]. Another example of pathways perturbed in cancer are programmed cell death (PCD) pathways[40]. In normal cells, DNA damage or other alterations that are typical to cancer often trigger PCD in order to avoid the loss of pathway regulation. However, in cancer, the acquired alterations disrupt PCD mechanisms in different ways such as promoting cell survival (which is usually balanced against by PCD), silencing tumor suppressors like *p53* or *APC* or by over-expressing anti-apoptotic genes.

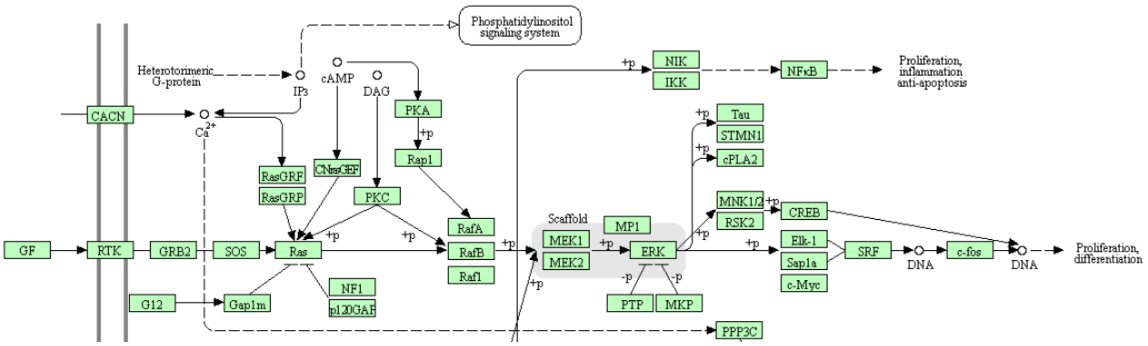


Figure 3: A schematic view of a pathway. The pathway shown here is the classical MAP kinase pathway from KEGG[41]. The cascade usually starts with the activation of the receptor tyrosine kinase (RTK) and ends with the initiation of the proliferation process. A green box represents a gene product. A circle is a metabolic compound. A solid line represents molecular interaction. Dotted lines represent indirect interactions or unknown interactions. A flat arrow (⊣) represents inhibition. An arrow (→) indicates activation or a product. +p denotes phosphorylation reaction. White boxes and open text represent other pathways or cellular process. Source: The KEGG website[42].

1.3.2 Main pathway databases

1.3.2.1 Reactome

Reactome[43] is the largest freely available source for pathways in human, containing 2244 pathways. The basic unit of Reactome is the ***reaction***. A reaction is any event that converts inputs to outputs, where inputs and outputs are physical entities such as small molecules,

proteins, lipids or nucleotides, or complexes of these. Reactions are grouped into pathways that take into account their interdependencies. Pathways can nest, i.e., they can have other pathways as entities. In addition to human pathways, which are the emphasis of Reactome, pathways for other organisms are computationally inferred using orthologs of human proteins. Reactome pathways are manually annotated by experts: the editors select a series of topics to annotate, and then invite bench biologists to author database "modules". Subsequently, full time curators ensure that the modules are complete and internally consistent. After curation, the module appears on a private website for inspection by peer reviewers and becomes publicly available afterwards.

1.3.2.2 KEGG

KEGG[41] (Kyoto Encyclopedia of Genes and Genomes) is a knowledgebase for diverse biological data including genes, compounds, drugs, and reactions, and is one of the popular databases for pathways. It is not restricted to human data and contains information for more than 6000 organisms. While KEGG contains references to other databases, it is intended to be self-sufficient by internally deriving all the biological knowledge needed including genes, reactions and molecules of the pathways they construct. The KEGG project was initiated in 1995 in Japan[44] and currently contains 530 pathways in human. However, it is not totally freely available as of 2011, and today only a subset of the pathways can be freely used[45].

1.3.2.3 NCI PID

The Pathway Interaction Database (PID)[46] is a collection of curated and peer-reviewed pathways, created in a collaboration between the US National Cancer Institute (NCI) and the Nature publishing group. It focuses on regulatory and signaling pathways, and is mainly intended to facilitate cancer and other regulatory biology research.

PID addresses two issues with today's representation of biological pathways: (1) pathway boundaries are often fuzzy. That is, two scientists may include different interactions in the same pathway, since pathways may vary under different conditions. In addition, we cannot guarantee that all scientists mean exactly the same thing when they address the same pathway. In the absence of ground truth, PID views pathways as abstract processes, allowing them to be dynamically adjusted (i.e. including or excluding interactions for a specific pathway) according to the user. (2) The levels of detail by which pathways are represented are not always the same. For example, one might want to represent an entire cascade or a pathway using a single entity in order to reduce the complexity of the pathway. In other cases, we might have different stringencies for including or excluding unreliable or partially unknown

information. PID tries to solve this in a similar manner, by allowing the user to customize the processes according to his or her needs.

PID is a subset of the "NCI-Nature Curated" collection of pathways, and chosen pathways emphasize potential drug targets, suggestions made by users and reviewers, and other mechanisms known to be of interest to the cell signaling community. All 212 pathways in PID are freely available.

2. Computational background

2.1 Steiner Trees

2.1.1 The Steiner Tree problem

The Steiner tree problem is the basis for a class of problems in graph theory. Shared among the many different settings of the problem is the objective to find a connected subnetwork (or several connected components) of a background network, while optimizing a function quantifying the extent of connectivity among a set of predefined nodes. The difference between this class of problems and the minimal spanning tree (MST) problem is that in the Steiner tree problem we are required to find an optimal subgraph connecting a subgroup of the nodes, rather than all of the nodes in the graph, while allowing nodes not in the subgroup to serve as intermediates.

We first introduce the most simple problem, called the *Steiner tree problem* (**Figure 4**): the input is an undirected graph $G = (V, E, W)$, where $W: E \to R_+$ is a positive weight function on the edges and a set of predefined vertices $V' \subseteq V$ called *terminals*. The objective is to find a connected subgraph $T = (V_T, E_T)$ such that $V' \subseteq V_T, E_T \subseteq E$ and the weight of T, defined as : $W(T) = \sum_{(u,v)\in E_T} W(u, v)$ is minimized. Note that T must be a tree and all its leaves are in V', otherwise it is easy to show that T is not optimal. All intermediate nodes in T (i.e. the nodes in $V_T \backslash V'$) are called *Steiner nodes*. This setting generalizes to the "Steiner forest" problem by allowing T to contain more than one connected component. Steiner tree problems can be defined on undirected or directed graphs. Our focus will be of problems for undirected graphs.

The Steiner tree problem was shown to be NP-complete[47] by reduction from the *exact 3-set cover* problem. The best polynomial approximation algorithm shown to the problem gives a 1.39-approximation[48].

In this work we will focus on the "Prize Collecting Steiner Tree" variant.

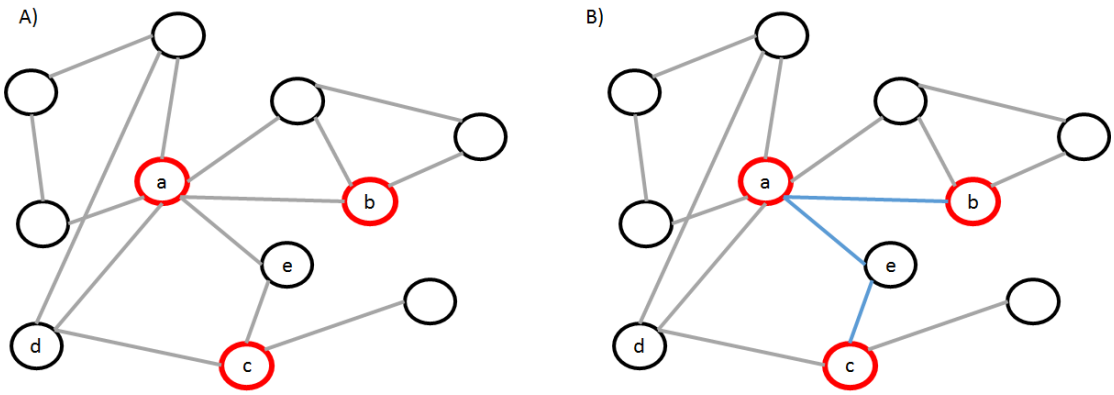


Figure 4: Example of the Steiner Tree problem. A) The input graph to the problem. The red nodes a,b,c are the terminals we wish to include in the tree; all other nodes can be Steiner nodes. In this example all edges are of constant weight. B) An optimal solution to the problem, including a,b,e,c. Note that the tree on a,b,c,d is also optimal.

2.1.2 The Prize collecting Steiner Tree (PCST) problem

In this problem (**Figure 5**) the input is an undirected graph $G = (V, E, W, P)$. W is a positive weight function on the edges as before and $P: V \rightarrow R$ is a weight function on the nodes. The objective is to find a subtree maximizing the sum of node weights minus the cost of edges in the subtree. In this formulation nodes with positive weights are called *prize nodes* and all other nodes are called *Steiner nodes*. Formally, the objective is to find a subtree T of G that maximizes:

$$(1)\quad \text{Score}\,(T) = \sum_{v \in V_T} P(v) - \sum_{(u,v)\in E_T} W(u,v)$$

Hence, instead of having a fixed set of terminals that must all be connected by the tree, here all prize nodes are predefined terminals, and some of them may not be included in the optimal subtree if connecting them is too expensive.

The PCST problem is NP-hard, as can be seen by a simple reduction from the Steiner tree problem. The best known polynomial approximation guarantee to the problem is 1.967[49].

Variants of the Steiner tree problem were applied before to biological problems in order to uncover altered mechanisms and pathways, notably by E. Fraenkel's group: Huang and Fraenkel[50] applied a branch-and-cut algorithm for the PCST problem[51] on transcriptomic, phosphoproteomic and genetic screen data to detect changes in regulatory and signaling pathways in yeast. Bailly-Bechet et al.[52] introduced a message-passing based algorithm (called

msgsteiner) to solve the PCST problem and applied it on transcriptomic data related to pheromone response in yeast. Tuncbag et al.[53] used msgsteiner and expanded its objective to a prize-collecting Steiner forest (PCSF) formulation (see below), in order to discover multiple altered pathways that are induced by pheromone response in yeast using transcriptomic and proteomic data. Gitter et al.[54] generalized the PCSF problem for a cohort of patients in order to find a "consensus network" that is shared across multiple patients, by introducing artificial prize nodes that reflect the node's inclusion frequencies among the individual networks.

Here, we are interested in the Rooted PCST (RPCST) problem. RPCST has the same objective as in (1) but in this setting, we restrict the tree T to contain a single predefined node called the *root*.

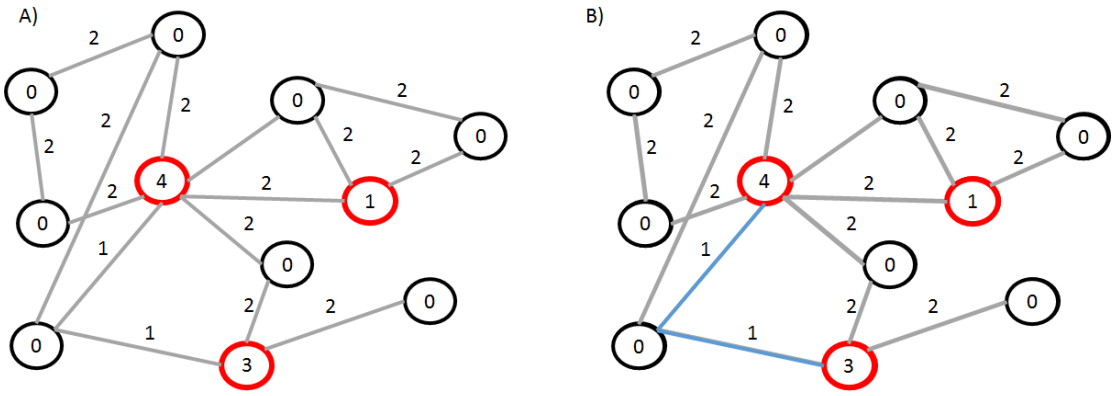


Figure 5: The Prize Collecting Steiner Tree problem. A) An input graph to the problem. Numbers in nodes represent the prize function P. Red nodes indicate nodes with positive prizes while the rest are Steiner nodes. Numbers on edges represent the weight function on edges W. B) The subtree of the blue edges is the unique optimal solution and has score 5.

2.1.3 Prize collecting Steiner Forest

The Prize collecting Steiner Forest (PCSF) problem is a natural extension of the PCST, allowing more than one tree in the solution while regulating the number of connected components. Mathematically, the objective is to find a forest T that minimizes:

$$(2)\quad \sum_{v \notin V_T} P(v) + \sum_{(u,v)\in E_T} W(u,v) + \omega k_T$$

Here k_T is the number of trees in the final solution and ω is a parameter that regulates the number of trees. The forest needs not span all prize nodes in the graph. Note that if we remove ωk_T from the equation and restrict T to be a tree, the objective is equivalent to (1). We will now discuss the relations between RPCST and PCSF.

2.1.4 Reducing PCSF to RPCST

There is an easy and practical reduction from a PCSF to RPCST. The reduction works as follows (**Figure 6**): Given an input $G = (V, E, W, P)$ and ω to the PCSF problem, we create a new graph $G' = (V', E', W', P)$ such that:

$V' = V \cup root$, where $root$ is a new node that will play the role of the root.

$E' = E \cup \{(root, v) | P(v) > 0\}$

$\forall (root, v) \in E' : W'(root, v) = \omega$.

G' is the input of RPCST. It is easy to see that a solution T to the RPCST has a corresponding solution to the PCSF of same score, by removing the artificial root node and all its edges from T. Conversely, a solution of PCSF can be converted to a solution of RPCST by adding a root to the forest connected by a single edge of weight ω to a representative prize node from each component.

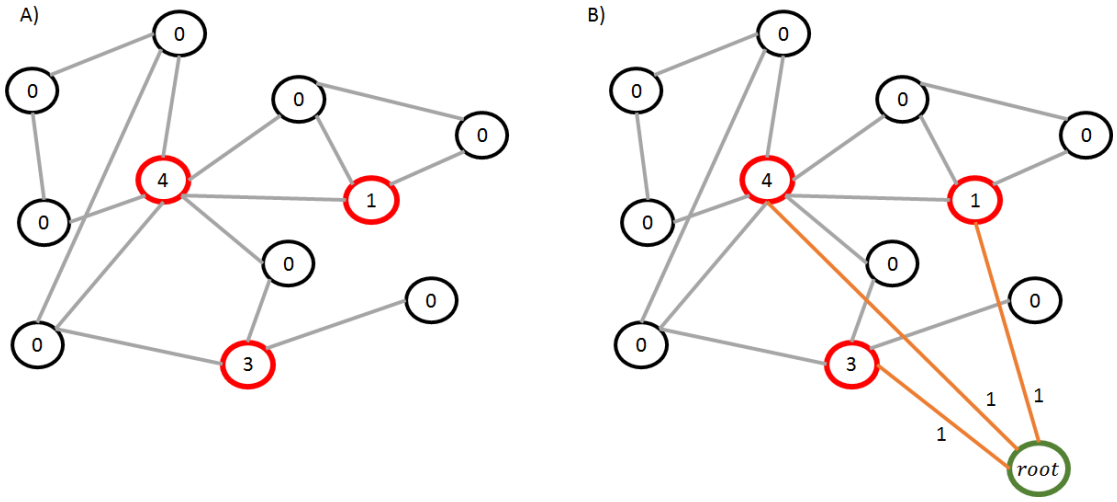


Figure 6: A reduction from PCSF to RPCST. A) An input graph of the PCSF problem. B) The reduction function introduces an artificial root node with edges of weight ω (in this example $\omega = 1$) to all prize nodes.

2.1.5 A Fast Prize-Collecting Steiner Forest heuristic

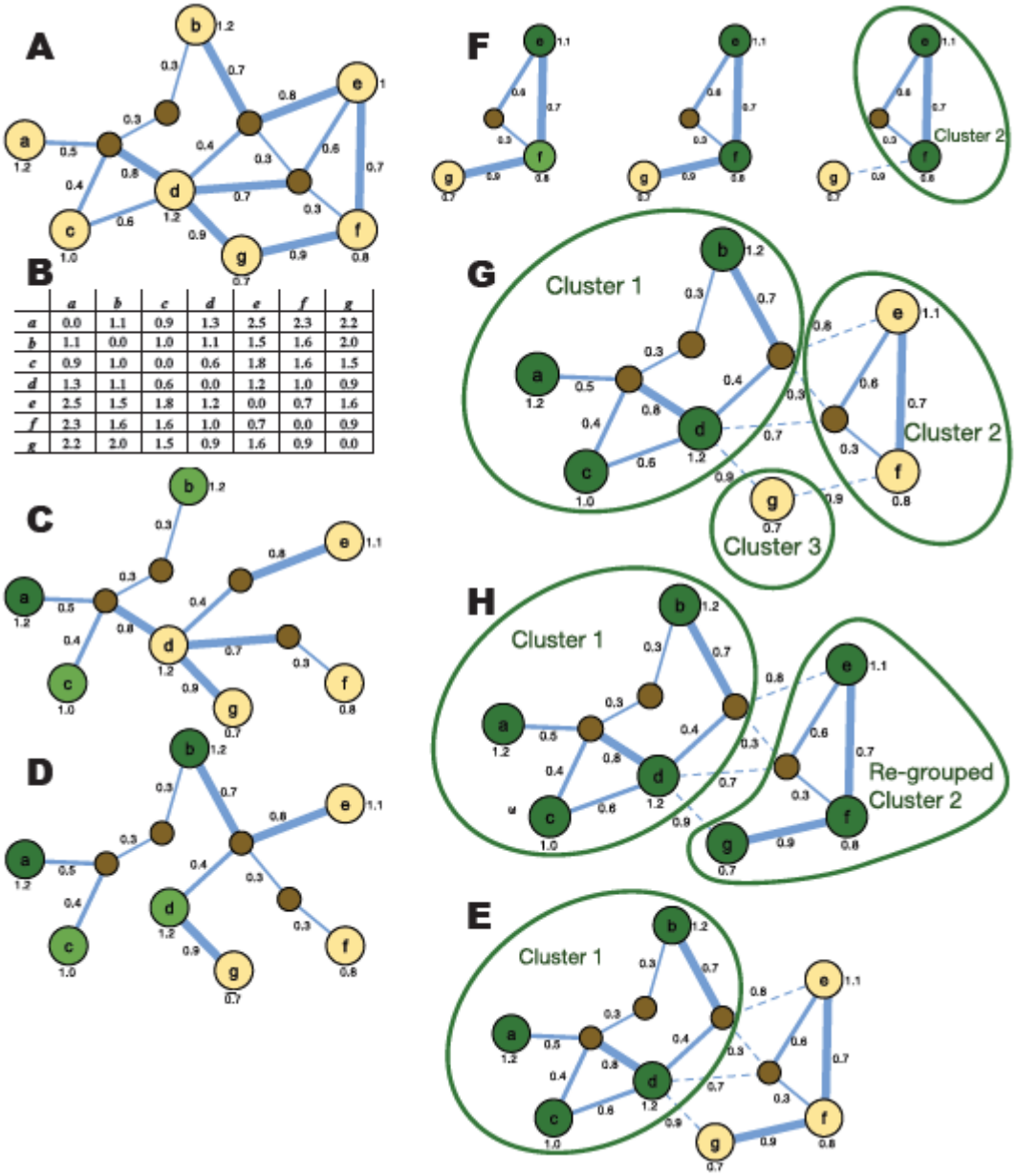
In this work, we use the implementation of Akhmedov[55], who introduced a fast heuristic to the PCSF problem. We will now describe Akhmedov's algorithm[56]. The input is a graph $G = (V, E, W, P)$ and a number $\omega \in R$. Nodes with positive prize function are termed *terminals* and all other nodes are called *Steiner nodes*.

The algorithm has two steps:

1. In the first step, the graph is divided into small clusters of high benefit in the following way (**Figure 7** gives a schematic description):
   Denote $i = 1$ and define all terminal nodes as "unassigned".
   1.1 First, we calculate the shortest path from every terminal to every other terminal in the graph, resulting in a matrix of shortest path distances D. Here the distance incorporates edge costs as well as possible costs for intermediate nodes. $D_{i,j}$ does not include the prize values of the terminals i and j.
   1.2 We randomly choose an unassigned terminal node v, assign it to Cluster i, and assign to that cluster every other terminal node u satisfying the clustering criterion: if $D_{u,v} < P(u)$ & $D_{u,v} < P(v)$, assign u to Cluster i.
   1.3 For every node $u \neq v$ assigned to Cluster i in the previous step: assign to Cluster i every *unassigned* terminal node t satisfying the clustering criterion with u.
   1.4 Increase i by one, repeat 1.2-1.3 until there are no unassigned nodes in the graph.
   1.5 Merge singleton and doubleton clusters with their nearest cluster: let G_k be a singleton or doubleton cluster and define the closest cluster to G_k as $G_{\min} = \underset{G_j}{\operatorname{argmin}} \sum_{v \in V_k} \sum_{u \in V_j} D_{v,u}$. Merge G_k and G_{min}.
   1.6 Return the final clustering of nodes.
2. In the second step, a final forest is computed as follows:
   2.1 For every cluster G_s from the previous step, we construct the complete subgraph $G'_s = (V'_S, E'_S, W')$: V_S' is the set of terminals in G_S, E'_S connects every two nodes in V_S' and for every pair of terminals $u,v \in V_S'$: $W'_{u,v} = D_{v,u}$. We also introduce an artificial root node and connect it with an edge to each terminal node such that for every terminal u: $W'(root, u) = \omega$ (see **Figure 8A and 8B** for a schematic view).
   2.2 Solve the minimum spanning tree (MST) problem on the resulting graph (**Figure 8B**).
   2.3 Prune all leaves for which the prize of the leaf is smaller than the cost of the edge to its parent in the MST.
   2.4 Exclude the artificial root node and return the resulting forest.

This heuristic was found to be satisfactory in quality for the purpose of our work and we incorporated it in our method by using a dedicated R package implementing the algorithm[55]. Other solvers for the PCST problem that we examined are based on belief propagation[52] or

ILP[57]. By and large, the latter methods were slightly more accurate but significantly slower, and hence we chose Ahmedov's heuristic.



	a	b	c	d	e	f	g
a	0.0	1.1	0.9	1.3	2.5	2.3	2.2
b	1.1	0.0	1.0	1.1	1.5	1.6	2.0
c	0.9	1.0	0.0	0.6	1.8	1.6	1.5
d	1.3	1.1	0.6	0.0	1.2	1.0	0.9
e	2.5	1.5	1.8	1.2	0.0	0.7	1.6
f	2.3	1.6	1.6	1.0	0.7	0.0	0.9
g	2.2	2.0	1.5	0.9	1.6	0.9	0.0

Figure 7: Clustering phase of PCSF: **(A**) The original underlying network. Terminal nodes are in yellow, Steiner nodes are in brown, edges are in blue and edge thickness corresponds to edge cost. **(B)** The matrix contains the shortest path length between every pair of terminals. **(C)** Choose a terminal at random (a here) and assign all terminals that satisfy the clustering criterion with it to its cluster (b, c here). **(D)** Iteratively for every terminal *v* assigned to the last cluster, find additional unassigned nodes that satisfy the clustering criterion with *v* and assign them to the cluster. **(E)** The final cluster is determined when no more terminals satisfy the criterion. **(F)** Repeat steps **C-E** until all nodes are clustered. **(G)** The resulting clustering may contain many singletons and doubletons. **(H)** Merge

singletons and doubletons with their nearest cluster and return the final clustering. Source: Akhmedov et al.[56]

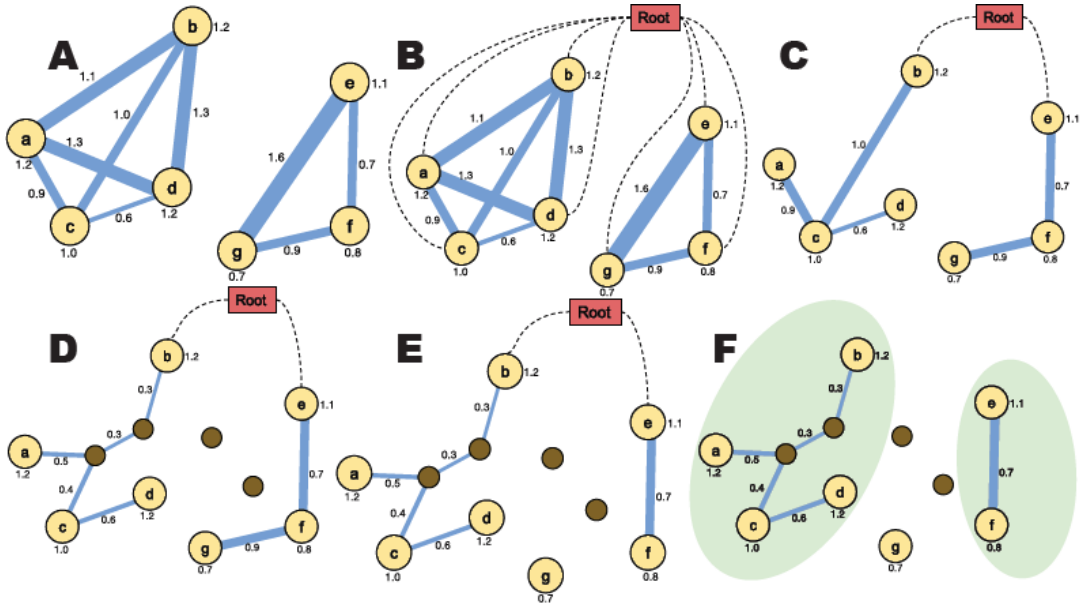


Figure 8: MST phase of PCSF**. (A)** Construct the complete cluster graphs from the terminal nodes of each cluster. Each edge is weighted as the length of shortest path between the nodes. **(B)** Add an artificial root node with an edge to every other node with cost ω. **(C)** Find an MST of this graph. **(D)** Collapse the shortest paths in the resulting MST, revealing intermediate Steiner nodes. **(E)** Prune all leaves for which the prize of the leaf is smaller than the cost of the edge to its parent. **(F)** Output the resulting forest. Source: Akhmedov et al.[56]

2.2 Cohort level methods for driver gene analysis

Computational research regarding driver genes first focused on distinguishing driver mutations from passengers in a cohort of patients (usually of the same tissue of origin): MuSiC[58] uses the statistical significance of higher than expected rate of mutations, along with pathway mutation rate and correlation with clinical features, to detect drivers. MutSigCV[59] estimates the background mutation rate of each gene and identifies mutations that significantly deviate from that rate. MEMo[60] tries to find small subnetworks of genes that belong to the same pathway and exhibit internal mutual exclusivity patterns. HotNet2[61] incorporates knowledge from PPI networks to find small connected subnetworks of higher than expected frequency of mutated genes using heat-diffusion process. TieDie[62] also incorporates PPIs and mRNA expression data to find overlapping subnetworks that possess

high degree of mutation and differential expression values using heat-diffusion. DriverNet[63] tries to find a parsimonious set of mutated genes that are linked to genes that experience deregulation of mRNA expression in a given PPI network. Many more methods for driver gene detection in cohorts are reviewed in Chang et al.[64] and Tokahim et al.[65].

We will now describe in more detail three methods that use different computational approaches to discover driver genes and driver-gene modules.

2.2.1 DriverNet

DriverNet[63] is one of the first cohort-level driver gene detection algorithms to incorporate both genomic aberrations and gene expression. It was derived based on observations from high-throughput datasets that suggested that driver mutations lead to abnormal gene expression of their interacting partners in the PPI network and in genes that share the same biological pathway. This phenomenon should be manifested in correlation between the true driver genes and the expression profiles of their associated genes in the PPI network. Moreover, these correlations should not be present for passenger mutations, thus differentiating drivers from passengers. The authors suggested that integrative analysis of genomic aberrations, transcriptional profiles and knowledge of the biological network would reveal driver genes through those correlations.

The input of the algorithm consists of three parts: (1) A matrix M with genes in rows and patients in columns. This is a binary matrix and $M(i,j) = 1$ iff gene i is mutated in patient j (mutation type could be any genomic aberration; here mutation is defined by SNV or CNV of the gene). (2) A gene expression matrix G with genes in rows and patients in columns. $G(i,j)$ stores the real valued expression of gene i in patient j. $G(i,j)$ is transformed to a binary matrix $G'(i,j)$ where $G'(i,j) = 1$ if gene i is differentially expressed (DE) in patient j as compared with a background normal population. (3) A gene network (also called the "influence graph") in the form of a square adjacency matrix I where $I(i,j) = 1$ iff j is directly reachable from i in the network (the network used in the paper was directed).

A schematic view of the algorithm is shown in **Figure 9.** The algorithm works as follows:

(1) A bipartite graph is built where nodes on the left represent genomic aberrations from M (green nodes show the genes that have a mutation in at least one patient) and nodes on the right are differentially expressed genes for each patient (g,p) from G' (for every patient, DE events are shown as red nodes). Edges are drawn under the

following conditions: for each patient p_k, draw an edge between node g_i in the left part and (g_j, p_k) on the right part, if g_i is mutated in p_k, g_j is DE in p_k, and g_i and g_j interact ($I(i,j) = 1$)

(2) In order to find the set of mutations that potentially explain the largest number of expression outliers, DriverNet applies a greedy algorithm that repeatedly selects the mutation (green node) with the highest degree and removes it along with its DE gene neighbors and the edges between them. If more than one mutation has the highest degree, one of them is chosen randomly. The algorithm stops when there are no uncovered DE genes left or after a predefined number of mutations were selected. Mutations are ranked by their degree. The algorithm is closely related to the minimum set cover problem, a well known NP-hard problem, and the solution described here is a known $\ln(n)$-approximation to the optimal solution (where n is number of genes).

(3) In order to assess the significance of the results achieved in step (2), a significance test based on $N = 500$ permutations is done: in each permutation, M and G' are shuffled randomly (e.g. assigning $M(i,j) = 1$ at random cells) while keeping the same number of total mutations and DE genes. The influence graph remains unchanged. Then, the statistical significance of a gene g chosen as a driver in some iteration of step (2) with COV_g DE neighbors is the fraction of times we observed a gene g' with larger DE neighborhood in the random sampling runs:

$$P - value(g) = \frac{\sum_{i=1}^{N} \sum_{j}^{S_i} I[COV_{g_{ij}} > COV_g]}{\sum_{i=1}^{N} S_i},$$

where S_i is the number of genes selected as drivers in sampling run i and g_{ij} is the gene chosen as driver in iteration j of sampling run i. P-values were FDR corrected for multiple hypotheses.

Finally, genes with significant outlier coverage (after FDR) are identified as drivers.

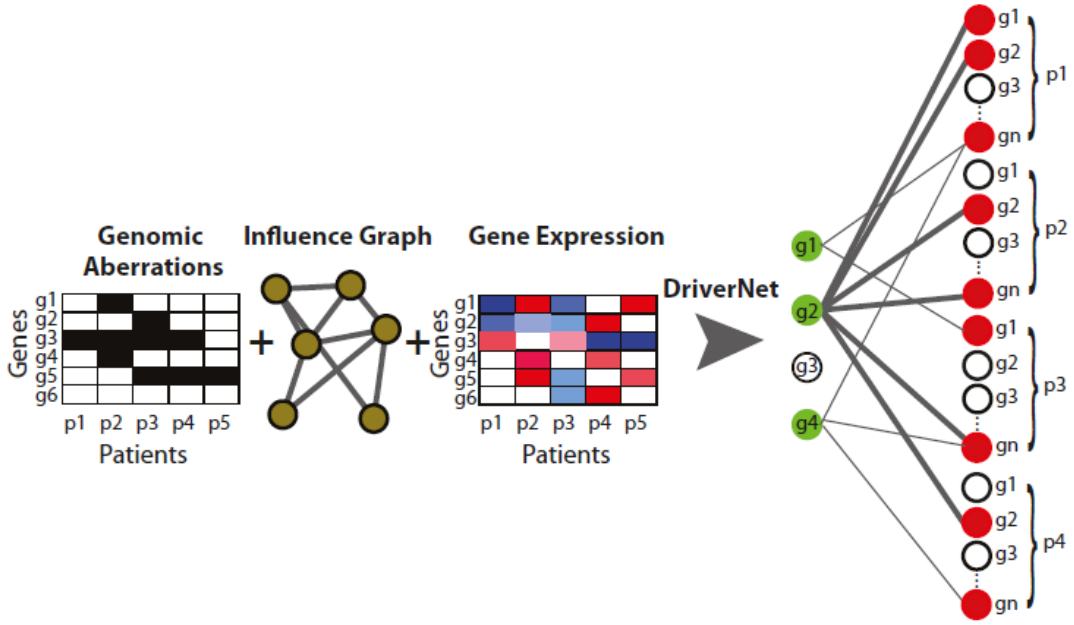


Figure 9: Schematic view of DriverNet. Given the genomic aberrations (left matrix), gene expression (right matrix) and influence graph, the bipartite graph shown on the right is constructed. Green nodes on the left part of the graph correspond to mutated genes and red nodes on the right represent expression outliers for each patient. Genes with the highest coverage of outliers (for example, g2) are nominated as drivers. Source: Bashashati et al.[63]

2.2.2 MEMo

Mutual Exclusivity Modules (MEMo)[60] is an algorithm for driver gene detection in cohorts that is built on the observation of mutual exclusivity between drivers in the same pathway. Experimental studies showed that different drivers have similar effects on the pathways they impact and that the number of such impacted pathways is lower than the number of drivers; hence the heterogeneity in driver genes across patients can be reduced by looking at the pathways they affect. Moreover, many tumor profiling projects have observed mutually exclusive genomic alterations across many patients. For example, in many patients either *TP53* is mutated or *MDM2* is copy number amplified, but only very few patients harbor both alterations[5]. As discussed before, a driver mutation grants the tumor cell selective advantage because of its impact on relevant pathways. In cases where a driver mutation is already perturbing a specific pathway, observations indicated that a second hit in the same pathway (leading to the same downstream effect) is far less likely to occur. Two biologically plausible scenarios may explain this mutual exclusivity pattern: (1) mutation in a second gene within the same pathway offers no further selective advantage and thus is not desirable. (2) Mutation

in the second gene within the same pathway actually leads to a disadvantage for the cell because of over-disruption to the pathway, and in the extreme case to cell death.

The authors define mutually exclusive driver gene sets according to three properties: First, member genes are altered (either via SNV or CNV) more frequently than expected by chance. Second, member genes are likely to participate in the same biological pathway or process. Third, genomic events within the network exhibit a statistically significant level of mutual exclusivity. The algorithm (**Figure 10**) is designed to identify mutual exclusivity models that answer these criteria and works as follows:

(1) In the first step, MEMo constructs a binary matrix M where rows correspond to genes and columns correspond to patients. It applies three filters to identify mutational events: the first filter identifies genes that are mutated significantly more than expected by chance using MutSig[59]. The second filter identifies copy number amplification or deletion as determined by GISTIC[66] or RAE[67]. The third filter identifies copy number altered genes with concordant mRNA expressions. That is, genes showing correlation between amplification/depletion and expression. Finally, we define $M(i,j) = 1$ iff gene i passed at least one of the three filters and the corresponding event occurred for patient j.

(2) This step identifies gene pairs that are in sufficient proximity in an underlying network (which MEMo terms as *similar pairs*). The proximity measure for a pair of genes (u, v) is the Jaccard Index (JI) of the sets of neighbors of the genes: denote $N(u), N(v)$ the set of neighbors of (u, v) in the PPI network. Then, $JI(u,v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$. Pairs with *JI* above a predefined threshold were identified as similar pairs.

(3) In Step 3, MEMo builds a new graph where nodes are genes and an edge connects every similar pair from Step 2. MEMo then extracts all maximal cliques from this graph.

(4) To achieve mutual exclusivity of mutations in each clique, MEMo adopts a greedy algorithm to keep only a subset of genes in each one: we define a gene as *informative* if the number of patients that harbor a mutation in it and in at least one other gene in the clique is smaller than the number of unique alterations (i.e. a gene is counted as the number of unique alterations it displays among all patients). For every clique received in step 3, the greedy algorithm starts by taking the most frequently mutated gene among the genes in the clique, and adds the next gene only if it is informative. Genes are examined according to descending mutational frequency among patients. Note that the resulting graph is a subclique that is not guaranteed to be completely

mutually exclusive (i.e. the group of patients with mutations in each gene may overlap). The score of the filtered clique is the fraction of patients that harbor at least one mutated gene from the clique. To assess the statistical significance of this score, MEMo uses a "switching permutation" algorithm[68] to generate random cliques of similar size for a given clique, while preserving the mutational frequency of each gene and the number of mutations per patients. For each clique, $N = 10{,}000$ random cliques are generated and the P-value of the observed clique's score is the fraction of random cliques with equal or higher fraction of patients altered in at least one clique member. MEMo also incorporates a method for testing sub-cliques for significance.

Finally, all participants of significant modules are identified as drivers.

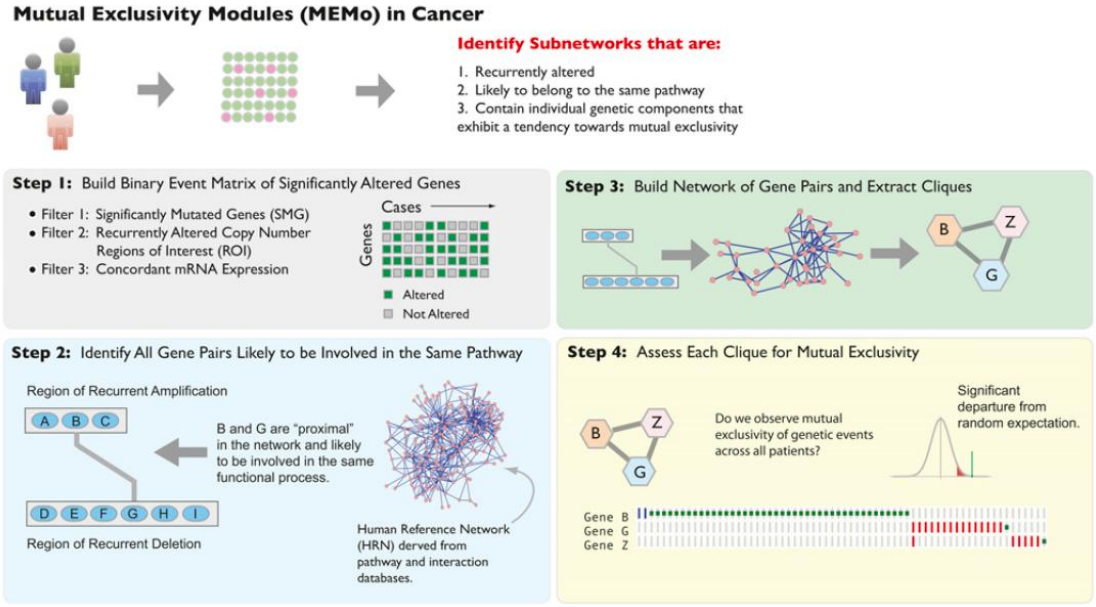


Figure 10: Schematic view of MEMo. Source: Ciriello et al.[60]

2.2.3 HotNet2

Although many driver genes have already been experimentally identified, there is huge heterogeneity in the frequency of driver mutations in the population. In other words, many driver genes are only altered in a small fraction of patients. This "long-tail" phenomenon complicates the efforts to identify driver genes, as rarely mutated cancer genes may be indistinguishable from genes containing sporadic passenger mutations. A different approach for cancer-related gene detection would be to exploit the fact that genes act together to construct biological pathways and complexes. That is, instead of identifying single genes, cancer genes can be inferred from a (small) *module* of genes that collectively form a biological role (like a complex or a pathway). HotNet2[61] aims to identify gene modules that are

significantly perturbed by somatic mutations using a directed heat-diffusion process. The input to the algorithm is a graph $G = (V, E)$ representing a PPI network and a vector h that represents the score of each node in the graph. Here h represented the mutation frequency of each gene in the population. The algorithm works as follows:

(1) Heat diffusion: Heat diffusion is a popular algorithm to estimate stationary states of high-dimension distributions, assuming dependencies between the random variables that are reflected through an underlying graph. At each step, nodes in the graph send and receive values (or *heat*) from their neighbors while retaining a fraction of their heat, governed by the parameter β. The process is run until a stationary state is reached. The diffusion process can be formulated in matrix form: The fractional amount of heat passed from node j to node i is given by the (*i,j*)-th entry of the square diffusion matrix *F* defined by:

$$F = \beta(I - (1 - \beta)W)^{-1}$$

Where W is the normalized adjacency matrix of the graph G:

$$W_{ij} = \begin{cases} \frac{1}{\deg(j)}, (j, i) \in E \\ 0, otherwize \end{cases}$$

Note that F depends only on the topology of G and not on the current heat vector h. To calculate the final heat each node absorbs in stationary state, we apply the heat vector on the diffusion matrix to obtain the exchanged heat matrix $\hat{E} = FD_h$ where D_h is the diagonal matrix with h on its diagonal. Taken together, F represents the fraction of heat passed between every pair of nodes for a single heat unit, and $\hat{E}$ represents exactly how much heat actually passes (according to the heat vector h).

(2) Identification of hot subnetworks. HotNet2 forms a weighted directed graph $H = (V_H, E_H)$ from the original graph G that is induced by the results of the diffusion process: $V_H = V$ and $E_H = \{(j, i) | \hat{E}(i, j) > \delta\}$. The rationale is that sufficient heat that passes between a pair of genes represents potential mechanistic connection between the genes that is manifested in the tumor. The algorithm then identifies strongly connected components in *H*.

(3) Statistical test for subnetworks. HotNet2 employs a statistical test to determine the significance of the number and size of the subnetworks determined in the previous step: the statistic is X_k, the number of strongly connected subnetworks of size $\geq k$ identified by HotNet2. To calculate the empirical null distribution of X_k, random permutations of the heat vector h are generated (while the graph remains

unchanged), and the algorithm is executed for each permutation to identify strongly connected subnetworks as described. The p-value of X_k is the fraction of permutations in which at least X_k components of size $\geq k$ were found. P-values are subsequently adjusted for FDR.

2.3 Personalized methods for driver gene analysis

The methods above focus on general driver gene detection, but do not aim to offer personalized means of diagnosis or treatment: individual patients may have different compositions of mutated driver genes (**Figure 11**). In addition, these methods rely on statistical power obtained by large cohorts and by doing so, they inevitably underestimate the importance of rare drivers that occur in only a handful of patients (this is known as the "long tail phenomenon"[69]) and are important only for them. Here we focus on patient specific driver gene prioritization.

Although many driver mutations were experimentally validated[3], personalized driver prioritization is needed for several reasons: 1) some patients carry mutations in dozens of known drivers (**Figure 11**). As discussed above, the number of active drivers in an individual is low (~7), hence it is essential to understand which are the true drivers for the individual. 2) Some patients do not possess mutations in any known driver (**Figure 11**), and for them one has to find putative drivers de novo. 3) Even if a patient has only few mutations in known drivers, and assuming they are all active, we still need to internally rank them since the number of therapies that can be given simultaneously to an individual is very low due to toxicity, adverse events, and cost.

Personalized driver gene profiles: To address the need for personalized driver gene identification and prioritization, one must develop methods that can operate on the data of a single patient. Several attempts have been made in this direction: DawnRank[70] uses a variant of Google's PageRank to rank an individual's mutated genes profile according to its effect on expression deregulation of downstream genes in a large directed PPI network. SCS[71] finds a parsimonious set of mutated genes that are sufficiently linked to downstream DEGs in a large directed PPI network. These methods rank putative driver genes for a patient. In contrast, Hitn'DRIVE[72] outputs a set of candidate driver genes without internal ranking. It tries to find a parsimonious set of mutations with short expected path lengths to a predefined fraction of

DEGs. The lack of ranking is a drawback from a treatment perspective, especially when the number of predicted genes is large. Lastly, PARADIGM[73] is a method that estimates the deregulated state of known pathways in a personalized manner, in order to infer which mechanisms are impacted by the canorous process. We will describe in detail DawnRank and SCS as they have similar objective to the method we developed here. We will also elaborate on PARADIGM since in our work we also try to infer cancer causes through its impact on known pathways.

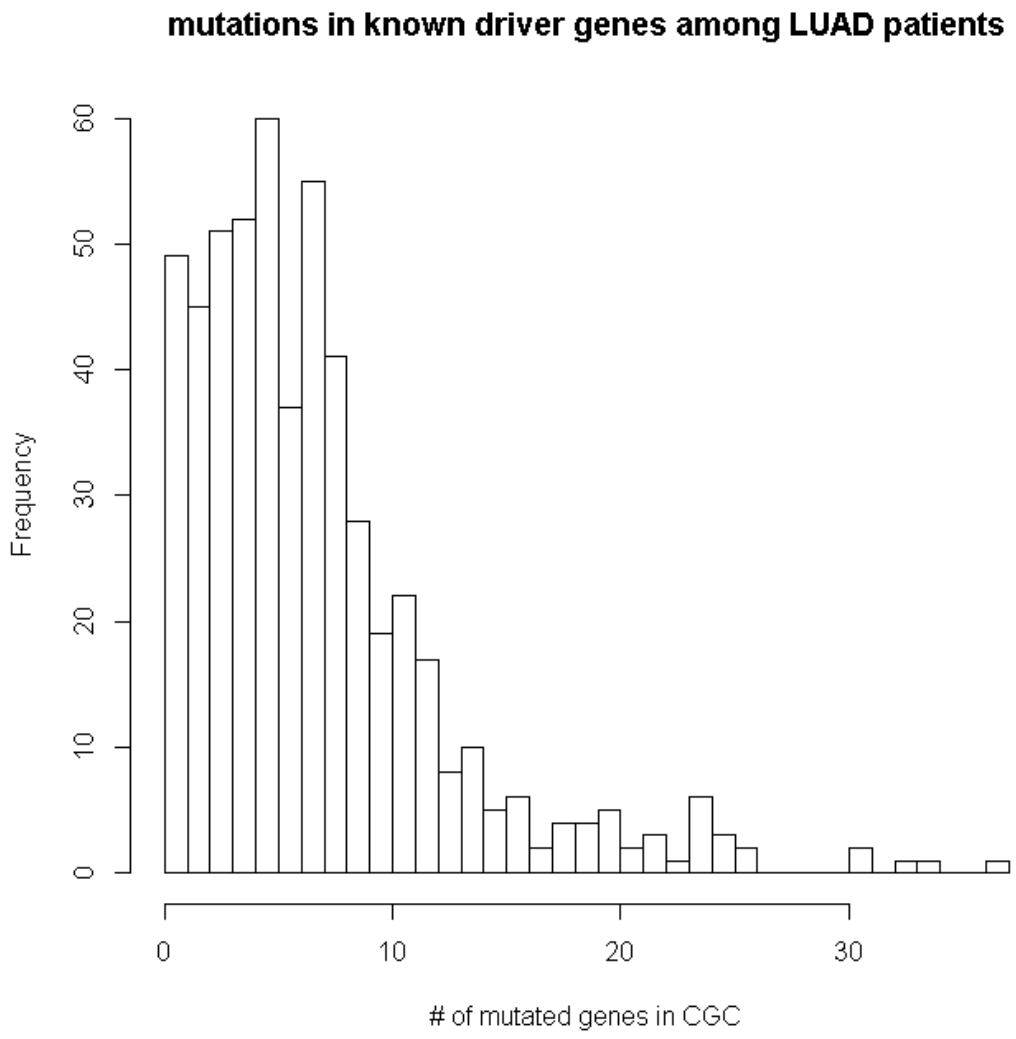


Figure 11: Distribution of the number of mutated genes that are known to be driver genes according to CGC (a curated source of known driver genes) per patient, among a cohort of 542 LUAD patients from TCGA.

2.3.1 DawnRank

DawnRank[70] is one of the first attempts to rank driver genes in a personalized fashion. The basic assumption of the algorithm is that driver genes affect deregulation in mRNA expression of downstream genes in the PPI network. The algorithm ranks the SNV and CNV profiles of an individual such that genes at the top of the ranking have a higher chance to be true drivers. The method is an adaptation of Google's PageRank algorithm[74] to the individual driver genes rankings problem: the input is a directed graph with N nodes represented by a binary adjacency matrix A, where $A_{ij} = 1$ iff (i,j) is an edge in the graph. The nodes represent genes and the edges represent PPIs. In addition, a vector of deregulated expressions f of size N such that $f_i = |\log_2 FC(g_i)|$ is given. $FC(g)$ is the fold-change in expression of the gene g in the tumor sample as compared to a matched normal sample. The *rank* of each gene is defined iteratively as:

$$(1)\ r_j^{t+1} = (1 - d_j) f_j + d_i \sum_{i=1}^{N} \frac{A_{ji} r_i^t}{\deg_i}, 1 \le j \le N$$

Where r is the ranking vector, t is the iteration index, $\deg_i$ is the in-degree of gene i and $0 \le d_j \le 1$ is the damping factor of gene j, representing the extent to which the ranking depends on the structure of the graph (higher d_i implies higher dependency on the graph). In DawnRank, the authors implemented a degree-dependent damping to avoid unstable rankings due to the "zero-one gap problem"[75] : $d_i = \frac{\deg_i}{\deg_i + \mu}$. μ was optimized using 100 random patients, by choosing μ that maximized the number of known drivers that were highly ranked (μ=3 was chosen here).

The algorithm is iterative: initially $r^0 = f$ and at every iteration t, r^t is updated according to (1). The algorithm stops upon convergence, i.e., when $\sum_{i=1}^{N} |r_i^t - r_i^{t-1}| < \epsilon$ (here ϵ = 0.001) or after 100 iterations if no convergence was reached. Note that the algorithm ranks *all* genes, not only the altered ones.

For individual ranking, only genes with at least one SNV or genes with altered copy number are ranked according to the final ranking vector r^n. DawnRank also generates a *cohort-level* ranking, by introducing a modified version of the Condorcet voting method[76]. In the Condorcet method, each voter ranks its favorite candidates as a full or partial list of predefined candidates. The Condorcet criterion determines the winner (called the *Condorcet winner*) as the candidate that would win every pairwise head-to-head competition against any other candidate. Because a Condorcet winner does not always exist due to possible circularity in pairwise competitions, a popular approximation to the Condorcet winner is to choose the

candidate that won the most internal pairwise wins across all voters (and if a Condorcet winner does exist, this method identifies it correctly). However, because the "lists" here are altered genes which are usually overwhelmingly partial (i.e. only a small fraction of the ~20k genes are mutated in every individual), there is a concern that frequently mutated genes will climb to the top of the ranking simply because many patients harbor them. To avoid this, the authors derived the following ranking method: for every patient i and every pair of genes (*A*,*B*), define:

$$PairwiseWinner(A,B) = \begin{cases} A, \delta(A) * Rank(A) > \delta(B) * Rank(B) \\ B, \quad Otherwise \end{cases}$$

$$\text{where } \delta(A) = \begin{cases} \delta, A\ is\ NOT\ mutated\ in\ patient\ i \\ 1, A\ is\ mutated\ in\ patient\ i \end{cases}.$$

In other words, we allow genes that are not mutated in an individual to win according to their derived rank, calibrated by the parameter δ that controls the tradeoff between preferring true mutations and avoiding bias from frequent mutations. In order to derive a cohort-level ranking, $PairwiseWinner$ is calculated for all possible pairs of genes in all patients. Genes are then ranked according to their total pairwise wins. The δ parameter was set to 0.85 after optimizing on 100 random patients.

2.3.2 SCS

Single-sample Controller Strategy (SCS) is a method for driver gene prioritization in individuals that uses network control theory. Network control theory considers how to choose the proper subset of network nodes to control the transition of the whole network from one state (e.g., normal state) to another (e.g., disease state). In our context, the network is a directed PPI network, the nodes represent genes and there are two states: normal and tumor. The state of the network is reflected by the mRNA expression of the genes. The objective is to find a parsimonious set of genes that control the transition of the network from normal state to tumor, which is reflected by the differential expression of mRNA in the tumor compared to normal samples. Here, control is manifested by connectivity: the goal is to find a small set of mutations that cover a maximal portion of DEGs.

The inputs to the algorithm are binary SNV and CNV vectors, mRNA profiles from matched tumor-normal samples for differential expression analysis and a directed PPI network. The algorithm works as follows:

(1) First, SCS calculates the fold change between the matched tumor and normal tissues. Every gene v for which $|\log_2 FC(v)| > 1$ is identified as DEG and is assigned +/- 1

value according to the direction of the fold change. Then, SCS calculates a personalized underlying network derived from the input network using the Random Walk with Restart (RWR) algorithm. RWR simulates a random walker's transition in the network from a starting node (or few starting nodes) with predefined starting probabilities. Given the starting probabilities, the probability that the random walker would reach a specific node in the network after $t + 1$ steps is given by:

$$p^{t+1} = (1 - r)Wp^t + rp^0$$

p^t is a vector in which the i-th element holds the probability that the walker would reach node i after t steps (we term i the *end node*). r is the restart probability (the probability that the walker returns to the starting point) and W is the column-normalized adjacency matrix of the graph. The algorithm iterates until it converges to a stationary state, and we interpret the values of p^n as the probability to reach every end node after an infinite number of steps. Assuming that there are k initial genes from which the walker could start with equal probability, the initial vector p^0 is defined as a vector with initial nodes having a probability of $1/k$ and the remaining nodes having a probability 0. The RWR function is solved using an iterative process and stops when $|p^{t+1} - p^t| < \epsilon$ ($\epsilon = 10^{-6}$ here). In SCS, the initial nodes are all the altered genes. To identify significant end nodes, a null distribution for end node probabilities is calculated using 100 random walks on 100 random networks with the same topological properties (i.e. same degree distribution). Then, a z-score is calculated for every end node as:

$$z_i = \frac{p_i - mean(SD_i)}{std(SD_i)}$$

where p_i is the probability of node i to be an end node in the original network (i.e. $p_i = p_i^n$). SD_i is the distribution of end node probabilities for node i generated using the random networks. Mean and std of SD_i are computed based on the 100 simulations. P-values for all genes are calculated using their z scores (assuming normal distributions) and the significant genes (P-value < 0.05, unadjusted) along with the altered genes and all the interactions between them construct the personalized network.

(2) Next, a greedy algorithm is used to find a parsimonious set of mutations that is linked to the set of DEGs:

2.1 Paths (termed *control paths*) from the set of mutations to the set of DEGs (also termed the *target genes*) are found using an iterative process: in the first

iteration, the group of target genes (denoted R_0) and all their inbound neighbors (denoted L_0) along with the edges between them are modeled as the directed bipartite graph $B_0 = (L_0,R_0)$. Then, a maximum matching in the graph is calculated and the set of nodes from L_0 that participate in the maximal matching constitute R_1 for the next iteration (L_1 are the set of inbound neighbors of R_1 as before). The process continues until $L_n = \phi$ (i.e. no inbound neighbors for R_n are left).

2.2 Using the paths obtained in the former step, a bipartite graph (M, D) is constructed such that *M* is the set of altered genes, *D* is the set of DEGs and there is an edge (*m*,*d*) if there is a path between the gene *m* and the DEG *d* in the subgraph obtained in step 1. Then, a parsimonious set of nodes from *M* that spans *D* is calculated using the standard approximation algorithm for minimum set cover.

2.3 In order to consider additional control paths between the mutations and the DEGs, SCS introduces a sampling method to construct *consensus models* for each mutation, i.e., a set of paths between the mutation and downstream DEGs, derived from 1000 sampling runs. At each run, some of the edges that were used to build the maximal matches are randomly selected to be replaced, i.e., they are removed from the matching and replaced by other edges to construct a new matching using the same algorithm. New control paths are then calculated according to 2.1. Finally, a new set cover is recalculated according to the new set of control paths. Each consensus module is weighted according to the frequency of its edges in the sampling runs. The driver gene ranking is then calculated as a ranked list of genes according to the total weights of their corresponding consensus models. This ranking is the output of the algorithm.

2.3.3 PARADIGM

PARADIGM[73] is an algorithm for personalized analysis of cancer patients that tries to infer the deregulated state of curated pathways in order to decipher the mechanisms altered by the cancerous process in each individual. PARADIGM models a pathway as a probabilistic graph: CNV and mRNA expression serve as random variables for every gene in the pathway and the pathway interactions model the dependencies between genes. Then, the deregulation state is estimated from the joint probability of the state of genes that participate in the pathway, as reflected by their CNV and expression values.

PARADIGM models the pathway as a factor graph. Factor graphs are graphical models that aim to learn probabilities on networks from observational data (CNV and expression in our case) and known dependencies (as modeled by the graph). Factor graphs are constructed as bipartite graphs that contain two types of nodes (**Figure 12**): variable nodes, which represent observational entities and factor nodes (or factors), which represent functions of those entities. The edges in the graph complete the model: they connect the variables upon which the function (represented by the factor) is built. Taken together, every factor node represents a function of its neighbor variable nodes, and the value of an entity (represented by its variable node) depends on all the functions it participates in (represented by its neighbor factor nodes).

In our case, each gene G in the network is represented by four entities: the copy number of the gene (G_{DNA}), its mRNA expression (G_{mRNA}), its protein level ($G_{protein}$) and its activity (G_{active}). Each entity can take on one of three values (which we term *states*): {1,0,-1} corresponding to activated, neutral or deactivated relative to control (e.g. as compared to normal tissue) respectively. The values may be interpreted differently depending on the type of entity. For example, an activated mRNA entity represents over-expression, while an activated copy number entity represents amplification of the gene. In addition, each edge in the graph has either positive or negative label according to the effect of the interaction.

The construction of the factor graph given a pathway is as follows (**Figure 13**): First, for every gene G, we draw edges with a positive label from G_{DNA} to G_{mRNA}, from G_{mRNA} to $G_{protein}$ and from $G_{protein}$ to G_{active}. Then, each interaction from the pathway is converted into a directed edge between the appropriate proteins according to the functional role of the interaction. For example, an inhibition of gene Y by gene X on the mRNA level would connect X_{active} to Y_{mRNA} (**Figure 13**). Each edge is labeled as positive or negative (corresponding to activation or inhibition).

The factors of this factor graph are defined as follows: for each entity x_i we add a factor $\phi(X_i)$ where $X_i = \{x_i\} \cup \{Parents(x_i)\}$ and $Parents(x_i)$ refers to all the nodes with edges into x_i in the directed graph (**Figure 13**). The function representing the factor is:

$$\phi(X_i) = \begin{cases} 1-\epsilon, x_i \text{ is in the expected state from } Parents(x_i) \\ \frac{\epsilon}{2}, otherwise \end{cases}$$

The expected state of x_i from $Parents(x_i)$ is a majority vote of the group's states: each parent px_j with a positive edge to x_i contributes $(+1) * (state(px_j))$ or $(-1) * (state(px_j))$ for negative edge.

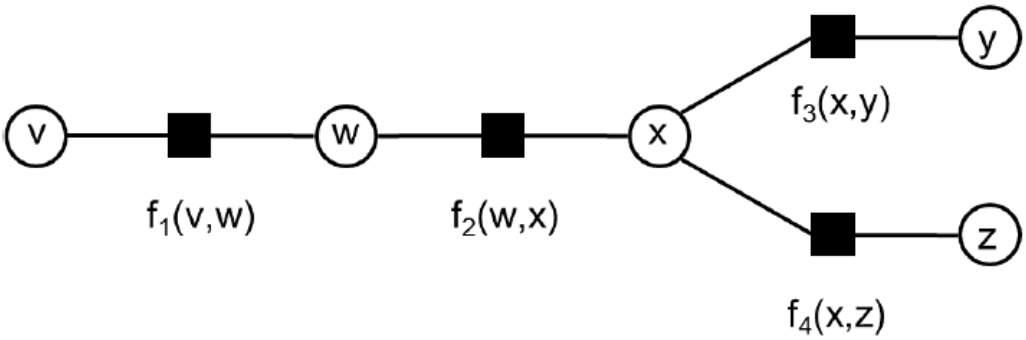


Figure 12: An example of a factor graph. Circles are variable nodes and squares are factor nodes. Edges connect factor nodes with the variables upon which their function is built. Likewise, the value of an entity is a function of all its factor neighbor's functions. Source: Rasmussen[77].

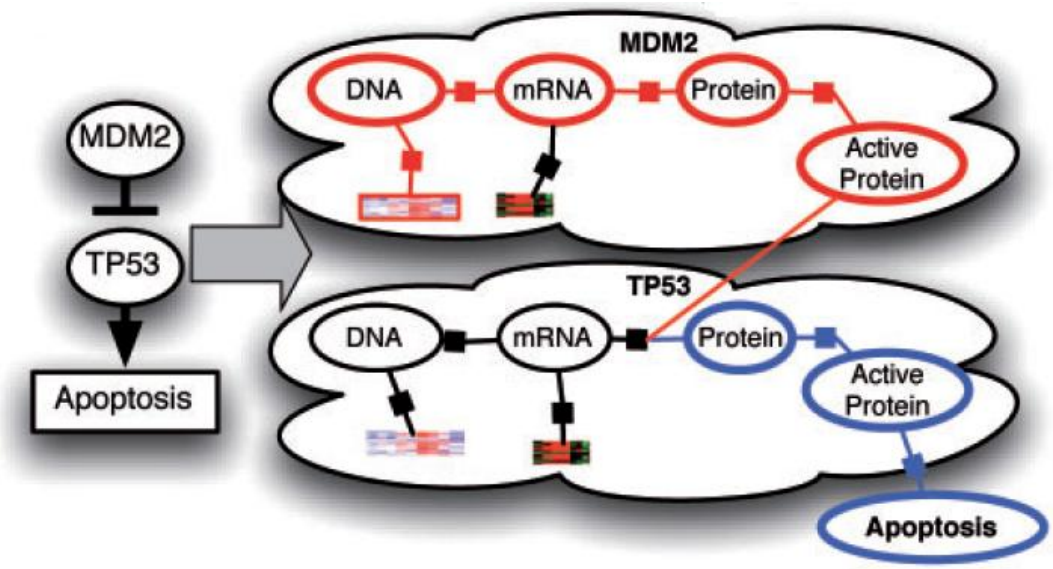


Figure 13: A factor graph constructed by PARADIGM. Each cloud contains the variables, edges and factors for one gene as explained above. In this example, *MDM2* is inhibiting *TP53* on the mRNA level, hence G_{active} of *MDM2* is connected to the factor that represents the activity state of *TP53*'s mRNA. In this example TP53 leads to apoptosis, which is also represented by a node to allow it to cross-talk with other pathways. Source: Vaske et al.[73]

Inference and parameter estimation: Let $D = \{x_1 = s_1, x_2 = s_2, \ldots, x_k = s_k\}$ be a complete assignment of variables for a patient. Let $\{S \subset_D X\}$ be the set of all possible assignments to a set of variables *X* that are consistent with the assignments in *D*, i.e., any observed variable x_i is fixed to its assignment in *D* while unobserved variables may vary. We estimate the prior probability of a hidden variable a, given the whole factor graph ϕ as follows: let $A_i(a)$ be the assignment $x_i = a$, then: $P(x_i = a|\phi) = \frac{1}{Z}\prod_{j=1}^{m}\sum_{S \subset_{A_i(a)} X_j}\phi_j(S)$, where ϕ_j is the j[th] factor and $Z = \prod_{j=1}^{m}\sum_{S \subset X_j}\phi_j(S)$ is the normalization constant. Similarly, the joint probability of

observing $x_i = a$ and D is: $P(x_i = a, D|\phi) = \frac{1}{Z}\prod_{j=1}^{m}\sum_{S\subset A_i(a)\cup D X_j}\phi_j(S)$. $P(x_i = a, D|\phi)$, $P(x_i = a|\phi)$ are learned using belief propagation, an algorithm for marginal distributions inference of unobserved variables using massage passing[78]. The estimation of the parameter ϵ was done using the expectation-maximization (EM) algorithm. Here, the parameters were learned for each pathway separately, by creating the described factor graph for each patient and applying the individual observations to estimate ϵ for each patient. All parameters estimated were then averaged over all patients to derive the final parameter. In order to estimate the final activity score for each gene i in the graph (represented by $G_{i_{active}}$), we calculate a likelihood-ratio based score to reflect the confidence that the activity state a of the gene i is consistent with the patient's data:

$$L(i,a) = \log\left(\frac{P(D,x_i=a|\phi)}{P(D,x_i\neq a|\phi)}\right) - \log\left(\frac{P(x_i = a|\phi)}{P(x_i \neq a|\phi)}\right) = \log\left(\frac{P(D|x_i=a,\phi)}{P(D|x_i\neq a,\phi)}\right)$$

We then calculate the integrated pathway activity (IPA) score of the gene i based on $L(i,a)$:

$$IPA(i) = \begin{cases} L(i,1), & L(i,1) > L(i,-1)\ and\ L(i,1) > L(i,0) \\ -L(i,-1), & L(i,-1) > L(i,1)\ and\ L(i,-1) > L(i,0) \\ 0, & otherwise \end{cases}$$

$IPA(i)$ reflects the activation/inactivation/neutrality of gene i in the context of the pathway (depending on the sign). Note that ϕ depends on the topology of the pathway, hence the same gene may have different likelihood-based and IPA scores for different pathways. Finally, by aggregating the IPA scores for all genes participating in the pathway, one can assemble a complete picture regarding the pathway's activity in the individual.

2.4 Centrality and centrality measures

Centrality is, in general, a property of a node or an edge in a network that aims to evaluate its role in the network. Most centrality applications tend to focus on network nodes. In this work we will study node centrality, which will be referred to from now on simply as centrality. The study of centrality stems from a simple observation of real-life networks: These networks are usually sparse and the properties of their nodes (e.g. their degree in the network or the average distance between them) are far from uniform. That is, some nodes have a more *central* location in the network compared to others and this central location might affect their role or importance in the network.

One popular example was given by Barabasi[79]: the internet is an alleged example of the ultimate freedom of speech. The content of a webpage or an article is hard to censor once published and everybody's voice can be heard with equal opportunity. If the internet were a

random network, this statement would have been true. But it is not. In order to be read you have to be visible; this visibility is measured by the number of links referring to your article (the more incoming links you have, the more visible you are). Given that the average webpage only has a handful of links to other pages, the likelihood that a random document links to your article is close to zero, leading to the unfortunate conclusion that this thesis (for example) is probably not very important in the sea of information that is out there… In contrast, major news sites (like CNN, New York Times or The Guardian) have much more incoming links for each article, making them very high degree nodes (also known as *hubs*) in the WWW network, and they are therefore considered more important. As this simple example demonstrates, it is interesting to explore different kinds of centrality measures and their implications to real life networks as a means for quantitative analysis of the role of the entities in those networks. Two very popular domains of interest in the context of centrality are human communication and social networks.

One of the first attempts to apply centrality concepts to human communication was done by Bavelas[80]. He hypothesized and showed that the centrality features in a small group of people influence group processes (for example the spread of a rumor). Another study by Leavitt[81] concluded that the centrality of individuals in social network determines behavior by limiting independence of action within a group, producing variability in activity and accuracy across different groups, which were associated with the leader of the group (the "central node"). Although some studies on the linkage between centrality and group behavior produced were inconsistent and gave contradictory results[82], others showed that centrality is relevant to the way groups get organized to solve at least some kinds of problems. Other applications of centrality that are reviewed by Freeman[83] include political integration and governance of countries, historically geopolitical importance of cities in transportation crossroads and efficiency in business corporations. In biological networks, Jeong et al.[84] observed strong correlation between the degree of a gene in the molecular interaction network and lethality of *Saccharomyces cerevisiae* cells when the gene was knocked down. We will cover more biological implications of centrality measures closer to our work later on.

One general intuitive theme that is shared across all studies is related specifically to node centrality and manifested in the following example: the node in the center of a star holds the most central position in the network, and is universally assumed to be structurally more important than any other node in this or any other network of similar size (and different topology). The challenge is to come up with a mathematical definition that will capture this observation. Attempts to address this challenge came up with three distinct properties that

are uniquely possessed by the center of the star: (1) it has the maximum possible *degree*; (2) it falls in the euclidean center *between* the largest possible number of other nodes and (3) since it is located at the minimum distance from all other nodes it is *closest* to them.

More generally, *degree, closeness* and *betweenness* are the major structural properties by which we define centrality and other measures are based on them[83].

Degree: This is the simplest centrality property of a node. In undirected networks, the degree of a node *v* is simply the number of *v*'s neighbors. In directed graphs, we usually split the degree of the node to in (out)-degree where *v* is the source (destination) of the edges. In order to compare degree values for nodes from different networks, there is a need to account for the different properties of those networks. The most prominent factor to consider is the network size. Hence, we may divide the degree of a node by the number of its potential neighbors (*n-1* for a network with *n* nodes).

Betweenness: The second property is based upon the frequency with which a node v falls within the shortest path between node pairs in the graph other than v. Mathematically: $Betweenness(v) = \frac{\sum_{i \neq v \neq j} \sigma(i,v,j)}{\sum_{i \neq v \neq j} \sigma(i,j)}$, where $\sigma(i,v,j)$ is the number of shortest paths from *i* to *j* that pass through *v* and $\sigma(i,j)$ is the total number of shortest paths from *i* to *j* (there could be many, of course). Edges could be weighted or unweighted. The betweenness has specific implications that are not achieved when only considering the degree. In telecommunication for example, a router that is located on multiple short communication paths linking other routers might maliciously modify incoming massages to manipulate the entire network. In contrast, a distant legitimate router with high degree may not pass on any communication. In other words, betweenness reflects the efficiency of the node's location in the network and thus its potential activity or even control over the network, not only its general connectivity. Note that the above formulation implicitly assumes that given that all shortest paths are of the same length, each can be selected with equal probability; otherwise this measure loses its interpretation as a measure of the node's efficiency in the graph. To account for network size, the normalization factor is the number of pairs we consider: $\binom{n-1}{2} = \frac{n^2-3n+2}{2}$.

Closeness: The third measure is based upon the degree to which a node is close to all other nodes in the graph. It is also related to the control of some nodes in the graph but in a different way: The closeness emphasizes the extent to which a node can *avoid* being controlled by other nodes. Bavelas[85] described non-central nodes as such that must rely on others for massage passing. Thus, the independence of a node is reflected by its closeness to all other nodes. Another intuition to the closeness importance is the "cost" of spreading massages across the entire network: If we need to choose a single node as the source of distribution, the node with

the highest closeness will give minimum distribution costs (assuming equal importance for all nodes).

Mathematically, $closeness(v) = \frac{1}{\sum d(v,i)}$ where $d(v,i)$ is the distance between *v* and *i* in the graph. If the graph is not connected than $closeness = \infty$ for all nodes since by definition every node *v* has at least one unreachable node *i* and so $d(v,i) = \infty$. The normalization factor is again (*n-1*).

3. PRODIGY

I this work, we develop a new algorithm for ranking of driver genes of an individual. The algorithm, called PRODIGY (Personalized Ranking Of DrIver Genes) scores mutations by their influence on deregulation of multiple known pathways. Unlike the methods described above, Prodigy collects multiple signals from many local views of the same tumor rather than one global view. These local views are based on curated pathways and each one reflects a different aspect of the deregulation state of the tumor. Thus, the extent to which a given mutation explains multiple pathway deregulations serves as a proxy to the likelihood that this mutation is indeed one of the drivers. Our algorithm assumes that driver mutations influence the deregulation of other genes in affected pathways. In particular the true drivers will have good connectivity to these pathways, and our method is designed to score such connections correctly using a variant of the prize collecting Steiner tree problem. By aggregating many local views for all mutations of an individual, a global picture can be made and the personalized landscape of drivers can be assembled and ranked.

In testing on five TCGA cancer cohorts spanning >2500 patients and comparison to validated driver genes, PRODIGY outperformed extant methods and ranking based on network centrality measures. Our results emphasize the pleiotropic effect of driver genes and show that PRODIGY is capable of identifying even very rare drivers. Hence, PRODIGY can assist oncologists in decisions regarding personalized treatment.

Caveats: Note that while we occasionally talk about driver mutations, all our analysis is done on the gene level and - as in SCS and DawnRank - different mutations in the same gene are not distinguished. Since the number of mutations per mutated gene in a patient is usually 1 (**STable 1**) this distinction is less important for personalized ranking than for cohort-level analyses. Also, as we shall see, often we identify and rank ten genes or more per patient, so the notion of drivers in this study is somewhat more lenient than is common in the literature. However, our results suggest that a larger number of predicted drivers actually contribute to the performance.

3.1 Methods

Given the set of mutated genes and the expression profile of an individual, we wish to rank the mutated genes in that individual. Our assumption is that the influence of driver genes is

disseminated along pathways and is manifested by DEGs. By aggregating evidence from multiple pathways for a mutated gene, we score the extent to which it explains the deregulation of the pathways. This score serves as a proxy to the likelihood that the gene is a driver in the patient. Mathematically, we score the influence of a mutation on a deregulated pathway using the undirected prize collecting Steiner tree (PCST) model.

The PCST model: In this problem (**Figure 14A**) the goal is to find in a weighted graph a subtree maximizing the sum of the weights of the nodes minus the cost of edges (see Computational background for details). In our context, edge weights are penalties reflecting PPI interaction reliability, positive node weights are prizes given to DEGs as they reflect the pathway deregulation that we want to capture, while other nodes that can serve as intermediate nodes in the tree (*Steiner nodes*) are assigned non-positive values serving as penalties. Given a node $g \in V$, the objective is to find a subtree T of G that contains g and maximizes:

$$\text{Score } (T) = \sum_{v \in V_T} P(v) - \sum_{(u,v)\in E_T} w(u,v)$$

In other words, the score of T is the total profit of pre-defined prizes minus the penalties of using intermediate edges and nodes. This model was shown to be suitable in different biological problems and in particular in scenarios where a mechanistic view is desirable[52,54].

Data and reference network: Prodigy uses two types of genomic data for each patient: the list of mutated genes, i.e. all genes with SNVs or small insertions/deletions in coding regions, and the profile of mRNA expression. mRNA expression profiles from healthy tissue samples are also utilized for differential expression analysis. Prodigy also uses two types of undirected interaction networks: 1) a global PPI network taken from STRING v10.5[86]. Here we used only physical interactions that were validated experimentally and interactions from other curated databases with confidence score > 0.7, so that only highly reliable interactions were included. The resulting network had 11,302 nodes and 273,210 edges. 2) A collection of pathways. Here we used either Reactome[87], NCI PID[46] or KEGG[41]. Information about the pathway databases used is given in **STable 2**.

The Prodigy algorithm

A schematic view of the algorithm is given in **Figure 14.** The algorithm works as follows:

Pre-processing Given a patient's mRNA expression profile (as read counts), differential expression analysis was done using DeSEQ2[88] by comparing the profile to a background

expression distributions from healthy samples of the same tissue of origin. All genes with > 2 log2-fold-change that are statistically significant (FDR = 0.05) were identified as DEGs.

The gene set of each pathway is tested for enrichment in DEGs using the hyper-geometric score, and pathways that are significantly enriched (FDR = 0.05) are called *deregulated*.

Driver - pathway scores We use a global interaction network $G = (V, E, W)$ where W is the edge confidence score. For a deregulated pathway p we also have its network $G_p = (V_p, E_p)$. Both networks are undirected. The influence score of the mutated gene g on pathway p is calculated as follows:

1. We construct a new network $G_{p,g} = (V_{p,g}, E_{p,g}, W_{p,g}, P_{p,g})$ that is derived from G, G_p and g, as follows: The nodes of the network are those of the deregulated pathway, g, and $N(V_p \cup g)$- their distance 1 neighbors in *G* :

$$V_{p,g} = V_p \cup g \cup N(V_p \cup g)$$

Its edges are those of the deregulated pathway plus all edges of the global network with both ends in $V_{p,g}$:

$E_{p,g} = E_p \cup \{(u,v) | u, v \in V_{p,g} \; and \; (u,v) \in E\}$,

The cost of the edges from p is 0.1. For the other edges, which originate from the global network G, their cost depends on their confidence score in that network, with edges of higher confidence costing less.

$$W_{p,g}(u,v) = \begin{cases} 0.1, (u,v) \in E_p \\ 1 - W(u,v), otherwise \end{cases}$$

Edges from the pathway are assigned a constant penalty of 0.1 since pathway databases do not provide confidence scores for the interactions, but those pathways are highly curated. In contrast, the confidence scores on the edges from the global network are given an upper bound of 0.8 so that their cost in $G_{p,g}$ is at least 0.2. The rationale is that we want to steer the algorithm to prefer the original pathway edges, while allowing some alterations.

Finally every DEG that belongs to the pathway has a positive (prize) score depending on its fold change (FC), and every other node v has a negative (penalty) score depending on its degree in $G_{p,g}$ as follows:

$$P_{p,g}(v) = \begin{cases} \log\big(FC(v)\big), v \in DEG \cap V_p \\ -degree(v)^{\alpha}, otherwise \end{cases}$$

Note that DEGs in $V_{p,g} \backslash V_p$ have negative values. The PCST problem aims to collect as much of the prize nodes value while paying least penalty for intermediate edges and nodes. Intermediate nodes that have high degree ("hubs") open more connection options and are thus penalized higher depending on their degree. The α parameter controls that penalty.

2. Having constructed $G_{p,g}$ we now seek a tree $T_{p,g}$ that contains g of optimal score. If Score($T_{p,g}$) ≤ 0 (i.e., no tree with positive score is found), an empty tree with score 0 is output instead.

3. To account for variability in pathway size and the number of DEGs in the pathway, the *influence score* of mutated gene g on pathway p is defined as the fraction of attained score out of the upper bound of all positive prizes in the pathway:

$$Infl(p,g) = \frac{Score\big(T_{p,g}\big)}{\sum_{v \in V_{p,g}} \max\{P_{p,g}(v), 0\}}$$

The overall *influence score* of g is $\boldsymbol{infl(g)} = \sum_{p \in DP} infl(p,g)$ where *DP* is the set of deregulated pathways of the patient.

Pathway filtering We compute driver-pathway influence scores for all mutated genes and all deregulated pathways. For the final score we exclude pathways for which more than half of the genes had a positive score. These are mainly very large pathways that have high connectivity in the global network, and therefore some genes may acquire positive influence scores by chance.

Gene filtering Genes that acquired positive scores in many pathways have greater chance to represent a true effect on the tumor than genes that attained positive scores for only few pathways, possibly due to the topology of the network. In some patients, when plotting the distribution of $Infl(g)$ scores across all mutated genes g (after filtering pathways), we observed a bimodal distribution (see **SFig. 1**). Typically, one distribution contains genes with high scores collected from many pathways and the other contains genes with low scores collected from a few pathways. We modeled this distribution as a mixture of two Gaussians and computed its maximum likelihood parameters using EM[89]. We then excluded all genes that had higher posterior probability to come from the distribution with the lower mean (**SFig. 1**). In case a bimodal distribution was not observed, we did not filter any gene.

Final ranking After the filtering steps, genes are ranked according to their overall influence scores.

Comparison to other methods:

We compared Prodigy to DawnRank[22] and SCS[23]. Since both DawnRank and SCS use directed graphs, the global PPI network used to test them was taken from the original publication. This network contained 11,648 nodes and 211,794 directed edges. To ensure that results are not derived primarily from the topology of the network, we also generated personalized rankings using three node centrality measures: node degree, closeness and betweenness (see Computational background for definitions). To produce rankings based on each measure, we calculated it on each of the networks $G_{p,g}$ and summed the results over all the networks for a final ranking.

Validation:

In order to validate rankings, we used a curated list of driver genes from the Cancer Gene Census (CGC) as gold standard. CGC is part of COSMIC[90], the largest database of somatic mutations in cancer. CGC contains mutations of different forms (gene amplifications, SNVs, translocations etc.) that were experimentally validated as driver mutations for different cancer types. Since we only used information about SNVs and short indels of each patient, we used as ground truth only genes that were classified by CGC as containing a driver SNV or indel (n = 248 out of 567). In this validation, we assumed that if a gold-standard gene was mutated in a patient, it is a true driver gene in the patient's tumor. We measured the quality of each method by means of precision, recall and F1 with respect to the gold standard (see **Supp**. **Methods**)

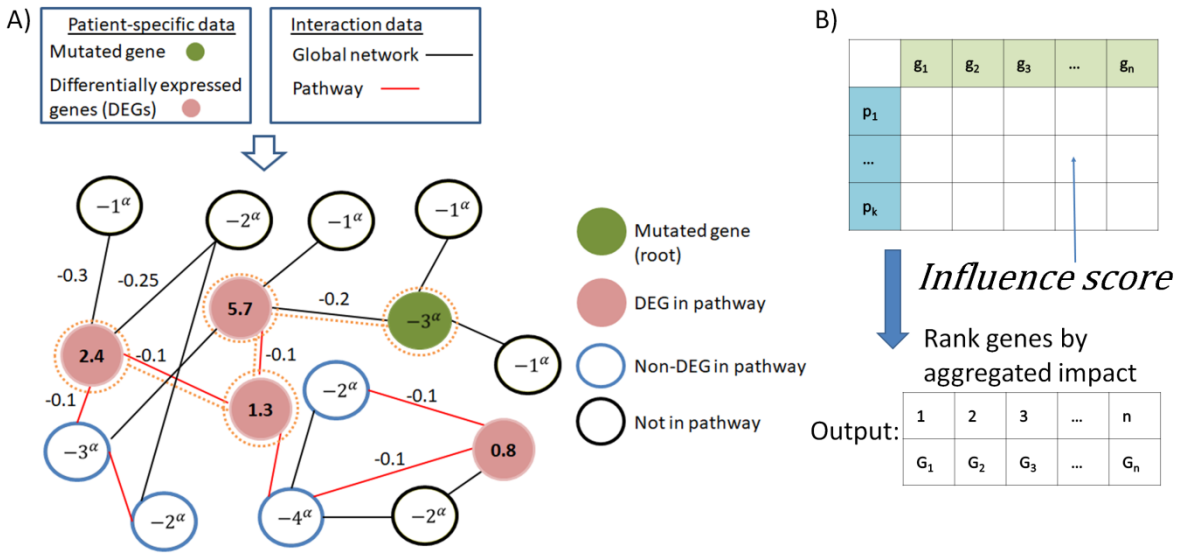


Figure 14: Outline of Prodigy's approach. A) Scoring the influence of the mutated gene *g* on the pathway *p*: The pathway and genes at distance 1 from it or from the mutated gene *g* in the global network, along with the global edges among them, constitute the network $G_{p,g}$ for analysis (see **Methods**). This is the network shown here. Node prizes (positive values) reflect the extent of differential expression of DEGs in *p*, and node penalties reflect other node's degrees (calibrated by the exponent α). Edge penalties reflect interaction confidence. The goal is to find a maximum weight subtree in the network rooted at the mutated gene *g*. Its weight is the score of the PCST solution. In this example the subtree marked by orange dotted lines is the PCST solution, of score 9-3$^{\alpha}$. The *influence score* of the pair (*p,g*) is the score of the PCST solution, divided by the sum of the values of DEGs that belong to *p* (10.2 here). B) After calculating the influence score for all pairs (*p,g*), we filter out some pathways and genes from the scoring matrix (see **Methods**). The final output is a ranking of the remaining genes by their aggregated score on the remaining pathways.

Driver-Pathway linkage:

Prodigy can quantify driver-pathway associations, allowing us to explore novel interactions and even cancer subtype-specific ones. Our hypothesis was that if driver gene g often deregulates pathway p then they will be observed together more frequently in patients of the cohort, and the deregulation state of p will be higher when g is acting as a driver. To test this conjecture, we focused on the ten top ranked genes for each individual and looked for driver-pathway pairs where the number of patients for whom the gene was ranked high and the pathway was deregulated was unexpectedly high according to the hyper-geometric distribution. For each pair, we then tested if p was more deregulated when g was classified as driver using t-test (see **Supp**. **Methods** and **SFig. 3** for more details).

3.2 Results

Driver gene ranking: We tested six ranking methods on 2569 samples from five cohorts of cancer patients from TCGA: COAD, LUAD, BRCA, HNSC and BLCA[91–95] (212, 487, 969, 502 and 399 samples, respectively). We used a training set comprised of 10% of the samples from each cohort to derive the optimal node degree weighting factor α in terms of F1, and used the chosen parameter to calculate personalized rankings for the remaining 90%. Prodigy's results were consistent across different α values (**SFig. 3**) with significant decline in performance for values > 0.2. $\alpha = 0.05$ was chosen for all cohorts.

Figure 15A shows the average precision, recall and F1 for Prodigy, DawnRank and for the three centrality measures using the Reactome pathways (see **Methods**). The results are reported as average values for the entire cohort as a function of the top N ranked genes. If an individual had less than N ranked genes, the last value for this patient was duplicated so that all quality measure vectors for all patients are of length N. Since SCS reported empty rankings for 720 samples (28%), it is not shown in **Figure 15A**. Performance of all methods on the set of patients for whom SCS produced results (the "SCS sub-cohort") is shown in **SFig. 5**, and performance for different cancer types is shown in **SFig.6-7**. Results for the KEGG and NCI pathway databases for the entire cohort were similar (**SFig. 4**).

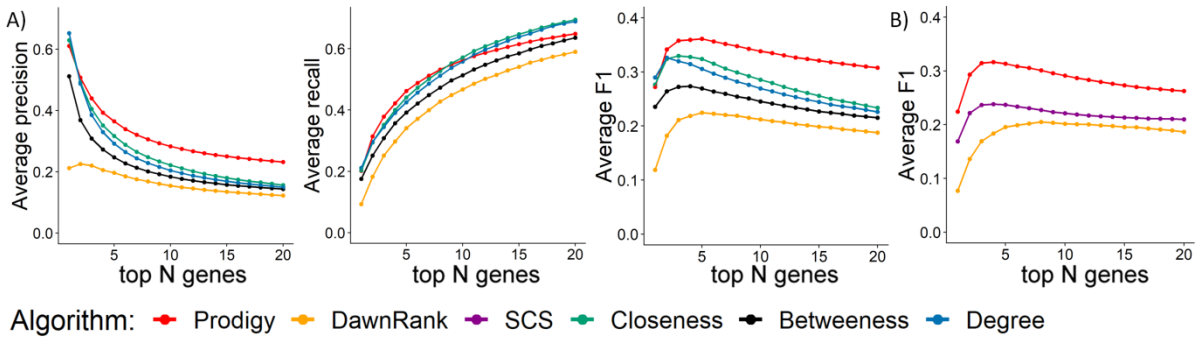


Figure 15: A. Average precision, recall and F1 across all patients (n=2340) as a function of the number of top ranked genes in the personalized profiles. Prodigy's results were derived using the STRING global PPI network (see **Methods**) and Reactome pathways B. Average F1 using the global network from the SCS and DawnRank papers, on those patients for whom SCS proposed drivers (n=1804).

Overall, Prodigy outperformed SCS and DawnRank in terms of F1, precision and recall. On the NCI pathways and for high values of N on KEGG pathways, SCS was better for the SCS sub-cohort. To ensure that the improvement in results does not stem from the difference in the underlying networks, we also tested Prodigy on the same network used by DawnRank and SCS

with two adjustments: (1) Since Prodigy works on undirected graphs, we ignored edge directions. (2) Since this network is unweighted, we gave weight=0.2 to all edges (and 0.1 to pathway edges as before, see **Methods**)**.** The results (**Figure 15B** and **SFig. 9**) clearly show that Prodigy outperforms DawnRank and SCS even on their network.

Remarkably, the centrality measures produced very good predictions, consistently better than DawnRank and SCS – but worse than Prodigy. These measures had better recall than Prodigy, probably due to the fact that no filtering was done on the centrality measures while Prodigy excluded genes not likely to be drivers for an individual. The fact that driver genes are associated with high network connectivity was previously discussed[72,96,97] and we observed it as well: in our global network derived from STRING, known drivers included in the CGC tended to have high degree and betweenness (**SFig. 7**). Our results emphasize the need to account for "hubness" property in methods for driver gene ranking. Prodigy accounts for this factor by penalizing Steiner nodes according to their degree. Taken together the results clearly demonstrate that Prodigy outperforms mere topology measures in capturing true driver genes.

Discovering rare drivers: One of the advantages of Prodigy is its ability to identify rare drivers, even when the gene is mutated in few patients. To demonstrate this ability we looked for mutated genes that had frequency < 2% in the cohort and were ranked in the top 10 drivers of individuals. The results are summarized in **Figure 16**. In some cohorts, most of the mutated genes were in fact rare (< 2%, **STable 3**), which is of course reflected in our results. On the other hand, Prodigy prioritized rare mutations to lesser extent than their frequency in the population (**STable 3**).

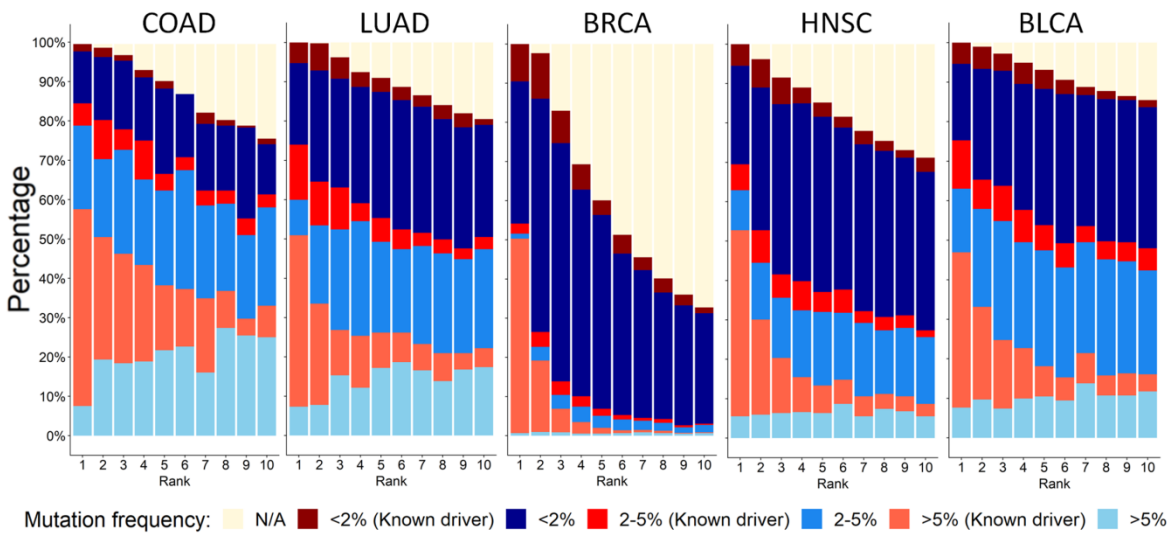


Figure 16: Prodigy discovers rare drivers. For each cancer type and for each individual we analyzed the top 10 genes according to the ranking. For k = 1,…,10 the plot shows the fraction of patients for whom the gene ranked k-th belongs to the respective frequency bin (as denoted by its color). N/A: patients for whom Prodigy ranked less than k mutations.

Prodigy was able to detect known rare drivers. For example, for the colon cancer patient *TCGA-AD-6899*, Prodigy ranked highly the gene *SRC*, a known driver in colon cancer. Remarkably, this patient was the only one (out of 212) who harbored a mutation in that gene. In HNSC, *FES* mutation was observed in five patients out of 502 (1%), and was highly ranked in two of them. *MTOR* was mutated in nine patients (1.7%) and highly ranked in five. *TSC2* was mutated in six patients (1.1%) and highly ranked in two. All of these genes are known HNSC-specific drivers according to CGC. In LUAD, *HIF1A* was mutated in two patients and was highly ranked in both. *RAD21* was mutated in eight (1.6%) and highly ranked in one. *ARAF* was mutated in five (1%) and highly ranked in one. *EED* was mutated in six and highly ranked in one. These are all known drivers of LUAD. The results show that Prodigy is capable of identifying even very rare drivers from the CGC. Taken together, we demonstrated Prodigy's ability to detect both rare and frequent drivers.

Driver gene-pathway linkage: We identified 1299 significant driver-pathway interactions (see **Suppl. File 1**). They include some very well-known interactions between *TP53* and sub pathways of the cell cycle in all cohorts except COAD and *TP53*-DNA repair pathways in the BLCA cohort. Moreover, the gene *A2M* was associated with "G alpha (i) signaling events" in the COAD, BRCA and BLCA cohorts. The G alpha (i) signaling pathway belongs to the GPCR family of signaling pathways, which are strongly linked to cancer[98]. This analysis can provide

new insights on the mechanism by which the drivers operate and can offer new targets for further research.

Multi-pathway effect: One of the main assumptions underlying Prodigy is that driver genes affect cellular process pathways, and therefore summarized scores from multiple pathways will improve our ability to identify them. This is in contrast to previous methods that took a global approach to driver gene prioritization based on a single unified picture of the state of the tumor[22,23]. In order to test whether multiple sources indeed contribute to the accuracy of prediction, we explored the performance as a function of the number k of allowed pathways per mutated gene. For k = 1,…,50, we used the top k scoring pathways of each gene for ranking and examined the average area under the precision-recall curve (AUPR) for each cohort (see **Supp. Methods**). **Figure 17A** shows that for all cohorts, AUPR improved with incorporating more pathways and plateaued at 15-30 pathways.

Since different pathways may partially overlap, we tested the extent of this overlap and its effect on performance. We computed the distribution of Jaccard Index between pairs in the top 20 scoring pathways of each gene (i.e., the number of genes that belong to both pathways divided by the number of genes in their union, **Supp. Methods**). The results show substantial overlap among the pathways that contribute to the rankings (**Figure 17B**). However, when we filtered out such overlapping pathways, assuming they contain the same information and thus unnecessary for accurate prediction, performance only moderately degraded (**Supp. Methods** and **Figure 17C**). Taken together, we demonstrated the usefulness of using multiple pathways in order to rank driver genes, even when there are overlaps among them.

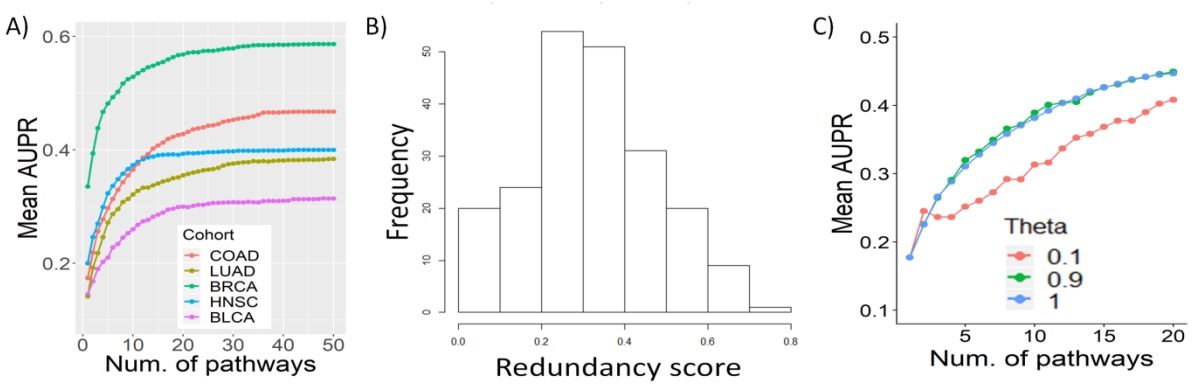
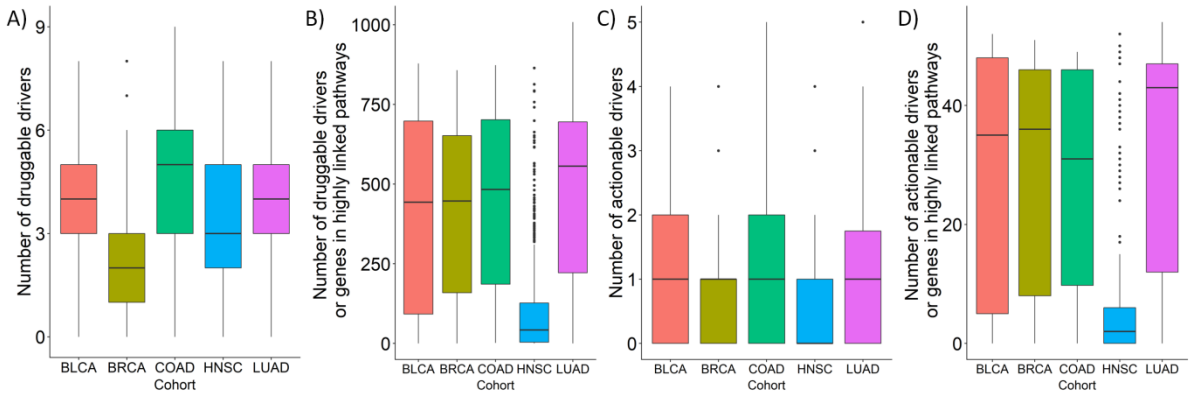


Figure 17: Multi-pathway effect: A) Mean AUPR as a function of the number of top scoring pathways per gene used to derive the results. B) The distribution of redundancy between the top 20 pathways per patient in the COAD cohort (n = 212, see **Supp. Methods**). C) Removal of pathway redundancy. The plot shows the AUPR for predicting driver genes in the COAD cohort when filtering out overlapping

pathways among the top scoring pathways per gene (**Supp. Methods**). θ is the maximum allowed Jaccard Index between included pathways ($\theta = 1$ implies no filtering).

Actionable and druggable targets: Prodigy's rankings can aid the oncologist in deciding on a patient's therapy, by matching treatment to the predicted driver genes. In order to explore this possibility we used two data sources: (1) DGIdb 3.0[99], which contains drug targets (or *druggable genes*, i.e., genes with directed pharmacotherapy). Here we used only cancer-specific sources from DGIdb and identified 1375 genes. (2) TARGET[100], which lists *actionable genes* (i.e., genes for which a genomic-driven therapy exists). The total number of actionable genes was 60. We explored not only the ranked mutated genes themselves but also the pathways that were highly linked (influence score > 0.8) to at least one gene of the top 10 ranked genes of an individual. The rationale is that these pathways are most altered by the driver genes and thus can be targeted in potential treatments. The results (**Figure 18)** indicate that most patients harbor at least one druggable driver (a druggable gene that was prioritized as a driver by Prodigy; mean: 3.32, sd: 2.01) but many do not contain any actionable drivers (mean: 0.82, sd: 0.87). As expected, the number of target genes increased substantially when genes from highly linked pathways were also considered. More importantly, the number of patients without any druggable or actionable gene decreased below 10%. The only exception was the HNSC cohort, where the number of patients without actionable genes remained high (35.8%) even when considering pathways. Hence, Prodigy is able to suggest possible therapeutic targets personally tailored to the patient's driver genes and uses information about the pathways that are deemed altered by the drivers in order to expand pharmacotherapy options.



	COAD	LUAD	BRCA	HNSC	BLCA
Num. of patients without druggable drivers (%)	4 (1.8)	21 (4.3)	155 (16.2)	38 (8.1)	8 (2)
Num. of patients without druggable drivers or druggable genes in highly linked pathways (%)	2 (0.9)	14 (2.8)	57 (5.9)	31 (6.6)	8 (2)
Num. of patients without actionable drivers (%)	83 (39.1)	172 (35.4)	507 (53.2)	278 (59.4)	155 (39.9)
Num. of patients without actionable drivers or actionable genes in highly linked pathways (%)	10 (4.7)	47 (9.6)	84 (8.8)	168 (35.8)	38 (9.7)

Figure 18: Actionable and druggable genes. The box plots show the distribution of the number of actionable and druggable genes (i.e. genes from TARGET[43] and DGIdb[42]) per patient across the different cohorts. A and C: The distribution of the number of druggable and actionable drivers among the 10 top genes ranked by Prodigy. B and D: The distribution of the number of druggable and actionable genes among predicted drivers and their highly linked pathways. The table describes the number of patients without any druggable/actionable genes in the four categories with respect to the cohort.

Implementation: Prodigy was implemented in R and the software is available in https://github.com/Shamir-Lab/PRODIGY. Mean runtime was about 5 minutes per patient on a 65 core, Intel(R) Xeon(R) 2.30GHz, 755GB RAM server.

4. Discussion

Personalized diagnosis of a cancer patient must precede the determination of a treatment plan. Deciphering the altered mechanisms and the mutations driving them gives a comprehensive picture of the state of the tumor and can be used to facilitate such diagnosis. Although many driver mutations were experimentally validated[36], the need to answer the personalized driver mutation prioritization problem is emphasized in different cases (**Figure 13**): 1) some patients harbor dozens of mutations in genes that are known to be drivers. The number of actual driver mutations per patient is believed to be no more than 7[1,4,8,9-11], hence there is a need to understand which are the true drivers for a specific individual. 2) Some patients do not possess mutations in any known driver gene. These patients probably harbor rare drivers and/or drivers that were not yet validated. In this case, we have no lead to the true drivers and it is essential to understand the state of the tumor in a personalized way. 3) Even if a patient has only a handful of mutations in known drivers (and assuming that they are all true drivers for him/her), there is still a need to internally rank them, since the number of therapies that can be given simultaneously to an individual is very limited due to toxicity[101], adverse events[102], and cost.

It is important to emphasize that this work was done on the gene level and it is blind to the specific properties of the mutations, i.e., different mutations in the same gene are not distinguished. This is a shortcoming, since it was observed that driver genes may harbor both driver and passenger mutations[6]. This is a limitation shared by all current methods for personalized driver gene analysis and most of the cohort-level ones. We observed that the number of mutations per mutated gene in a patient is usually 1 in the cohorts that we studied (**STable 1**), thus this distinction is less important in our case. We chose to focus on driver genes and not mutations for several reasons: (1) we cannot readily incorporate specific mutational data in the network formulation by which we score gene-pathway links (namely PCST). This is because PPIs are not variant-specific, so knowledge of the specific mutation in a gene will not change the PPI network as might be needed. (2) Although several specific mutations were validated as driver mutations and the driver gene landscape in coding regions is believed to be almost complet, identifying the exact driver mutations inside driver genes is a much more difficult task as the number of optional mutations in a single gene can reach thousands. In other words, the gold standard for driver mutations is smaller in orders of magnitude compared to driver genes. This makes validation less reliable, since we cannot confidently determine false positives/negatives in mutational resolution.

In this work, we introduced Prodigy- an algorithm for personalized prioritization of driver genes. The algorithm ranks mutated genes so that genes that are ranked at the top are more likely to be drivers for the specific patient. A physician can use Prodigy in order to tailor a personalized treatment in light of our prioritized genes. All current methods that tackle the same problem[22,23] use a global approach: they try to link mutated genes to transcriptionally altered genes in the tumor (i.e., DEGs) using a single large underlying PPI network. There is a limitation to this approach: large PPI networks are by definition not specific to condition or tissue and tend to be biased towards hubs and prone to large amount of errors[103,104]. The detection of mechanisms by which the driver gene operates is not less important than the identification of the drivers themselves since not all drivers have matched pharmacotherapy, and by detecting the processes affected by the driver we may have more treatment options. Taken together, using a single large PPI network to find drivers is somewhat contradictive to their mechanistic nature, which is more context-specific than holistic. We should note that the problem of finding biological processes de novo (e.g. a subnetwork that describes a biological mechanism) based on large cohorts went under extensive research (for example[61,105,106]), but it is much more challenging to so in a personalized manner where there is no statistical power. One way to improve the network is by filtering it according to the specific context (e.g. the tissue of origin), as was done computationally by Yeger-Lotem and colleagues[107]

While Prodigy also uses the global PPI network, it differs radically from extant methods in the way it combines that network with pathway-based information. Our approach computes the potential influence of mutated genes on each curated pathway separately, and constructs a final driver gene ranking from all these local views together. Our rationale is that the true driver genes will confer strong influence signals across multiple pathways and consequently climb to the top of the ranking. This approach directly addresses the limitation explained above: by using curated pathways we attenuate the non-specificity problem of the large PPI network and get a mechanistic explanation that is based on more reliable biological information.

Our results show that Prodigy is overall more accurate than extant methods in terms of precision, recall and F1. Remarkably, we show that naïve centrality measures perform better than all methods but ours. The relative success of the centrality measures probably stems from the effect of extensive research of known drivers, biasing them to very high connectivity in the PPI network (**SFig 10**) compared to other genes. This raises two major questions: (1)

How reliable are the results for methods that use large PPI networks, if they do not address this issue directly? Centrality measures are calculated regardless of the phenotype (differential mRNA expression in our case); hence if they perform consistently similar to ad-hoc methods, it might indicate that the results of these methods are artifacts of the underlying network. (2) What is the proper validation in the presence of such a pronounced confounder? In our case, using a list of gold standard drivers for validation is definitely not optimal because of their tendency to be central in the PPI network and because they do not guarantee any personalized causality of the putative drivers, a desirable property in driver gene identification. Given the limitations of driver mutations validation discussed above, this remains a challenge. Our analysis at least reassures that the results are not pure topological artifacts: Prodigy was superior over centrality measures in 11 out of the 15 scenarios presented (5 cohorts, 3 pathway sources) and far outperformed them when aggregating the results from all cohorts (**Figure 16, SFig 4, 6-8**). It should be stressed that since Prodigy does not utilize information regarding the tissue of the tumor, displaying the results according to the tissue of origin is somewhat artificial, and the aggregated results are a more appropriate validation to the nature of the method. While Prodigy ranks the genes without setting a cutoff for driver detection, our analysis shows the F1 scores peak around N=5. On the other hand, recall rises beyond N=5, so by using prior knowledge about driver genes and observing the actual influence scores of genes that are ranked lower, additional drivers can be pinpointed and used.

The effectiveness of our local approach is demonstrated by testing if using more pathways actually improves the results. **Figure 17** shows that it is indeed so. When deriving results based only on the top scoring pathways of every mutation, the AUPR improves as the number of pathways grows. In all cohorts, we observed that the number of pathways needed to reach a plateau in AUPR was ~15-30, which confirms our initial hypothesis that examining the tumor from multiple views helps the prioritization process. In this analysis we calculated AUPR and not precision, recall and F1 as before for two reasons: (1) AUPR gives a single value as opposed to the latter, which produce curves as functions of the top N genes. Since we wanted to demonstrate the quality of prediction as a function of the number of pathways, using AUPR was more suitable here. (2) For some methods, it was not always possible to calculate the AUPR in the former analysis. This is because they produce partial rankings (e.g. Prodigy only ranks some of the genes and filters out those that are unlikely to be drivers), and consequently a recall of 1 was not always possible to reach. In order to be able to calculate AUPR here, we did not apply any gene filtering in this analysis.

A major limitation of current methods for driver gene identification in large cohorts is their tendency to neglect rare drivers because they are overshadowed by frequent drivers. Here we showed that Prodigy is able to recover very rare drivers, mutated even in a single patient. Moreover, we showed that Prodigy detects drivers regardless of their frequency in the population, as corroborated by the fact that Prodigy identifies drivers in every frequency group (**Figure 16**) and by the fact that the number of rare drivers reported by Prodigy is smaller than their relative abundance in the population (**Supplementary table 3**).

We demonstrated that Prodigy can reveal linkage between a driver gene and pathways that are preferentially deregulated when the gene acts as a driver. The identified genes typically have multiple drug targets, and thus can suggest treatment decisions.

There are several limitations to our method. The main limitation shared by all personalized approaches (and by cohort level-methods by definition) including ours is that they cannot prove a causal link between the identified drivers and the tumorigenesis for an individual. This is due to the fact that all methods derive results for an individual based on a single sample from a single time point. More data are needed for causal identification of drivers. For example, matched normal-tumor samples from the same tissue of an individual might reveal strong transformation signal from normal to cancer cells when analyzing the differences in the mutational profiles and transcriptome. Another approach could be to analyze changes in the tumor over time using multiple samples from varying time points, with the aim to uncover the evolutionary path of the drivers. The local views approach that we developed here can serve as a basis for such future methods that will use more data to derive stronger results. Another problem is that the validation done here (and by former methods) is based on the CGC and thus limited, as described above. Additionally, although we emphasized our ability to make better use of pathways data, the PCST model that we use cannot reveal the exact mechanisms by which drivers affect pathways.

A technical problem in Prodigy is selecting the value of the parameter α. This value, which depends on the dataset, requires an easier optimization than the one done here. The role of α is to balance between the prizes of nodes representing DEG's and the hubness of intermediate nodes, which depends heavily on the topology of the underlying network. Here we optimized α using a training cohort and showed that Prodigy is robust for low values and for different networks (**Supplementary figure 3**). However, it might be needed to recalibrate α for other topologies. Finally, the running time of our method is slower than the other

methods, mainly because of the need to solve many PCST problems for each patient ($\#mutations * \#enriched\ pathway$ to be exact).

There are several future directions and improvements that could continue the work done here. This work provides proof of concept to the local approach for driver gene analysis, which could be utilized by other future methods. Other algorithmic approaches that are not based on the PCST model might find ways to quantify the mutation-pathway influence. In a similar regard, the calculation of the final score from the individual influence scores might be revisited, for example by weighing the pathways to reflect their different importance in a patient or by applying a machine-learning approach while viewing the influence scores as features. Another interesting direction would be to expand the driver-pathway linkage analysis and validation. Such analysis may have great potential for better understanding of the exact mechanisms affected by drivers. Finally, it should be highly useful to adapt our approach in settings where temporal data are used to decipher mechanistic changes due to the driver's impact. We demonstrated Prodigy's ability to pinpoint pathway-level alterations and the drivers potentially causing them, and by utilizing the time dimension one could have a fuller picture of the evolutionary process of tumorigenesis and the mutations driving it.

6. References

1. Stratton, M. R., Campbell, P. J. & Futreal, P. A. The cancer genome. *Nature* **458,** 719–724 (2009).
2. Carvalho, D. D. De *et al.* DNA Methylation Screening Identifies Driver Epigenetic Events of Cancer Cell Survival. *Cancer Cell* **21,** 655–667 (2012).
3. Futreal, P. A. *et al.* A census of human cancer genes. *Nat. Rev. Cancer* **4,** (2004).
4. Kim, H. & Kim, Y. M. Pan-cancer analysis of somatic mutations and transcriptomes reveals common functional gene clusters shared by multiple cancer types. *Sci. Rep.* **8,** 1–14 (2018).
5. Weinstein, J. N. *et al.* The Cancer Genome Atlas Pan-Cancer Analysis Project. *Nat. Genet.* **45,** 1113–1120 (2013).
6. Vogelstein, B. *et al.* Cancer genome landscapes. *Science (80-.).* **340,** 1546–1558 (2013).
7. Govindan, R. *et al.* Genomic Landscape of Non-Small Cell Lung Cancer in Smokers and Never-Smokers. *Cell* **150,** 1121–1134 (2012).

8. Tomasetti, C., Vogelstein, B. & Parmigiani, G. Half or more of the somatic mutations in cancers of self-renewing tissues originate prior to tumor initiation. *PNAS* **110,** 1999–2004 (2013).

9. Vogelstein, B. & Kinzler, K. W. The Path to Cancer- Three Strikes and You're Out. *N. Engl. J. Med.* **373,** 1893–1895 (2015).

10. Hoeijmakers, J. H. J. Genome maintenance mechanisms for preventing cancer. *Nature* **411,** 366–374 (2001).

11. Edinger, A. L. & Thompson, C. B. Death by design: Apoptosis, necrosis and autophagy. *Curr. Opin. Cell Biol.* **16,** 663–669 (2004).

12. Anna C. Schinzel, W. C. H. Oncogenic transformation and experimental models of human cancer. *Front. Biosci.* 71–84 (2008).

13. Kandoth, C. *et al.* Mutational landscape and significance across 12 major cancer types. *Nature* **502,** 333–339 (2013).

14. Xie, M. *et al.* Age-related mutations associated with clonal hematopoietic expansion and malignancies. *Nat. Med.* **20,** 1472–1478 (2014).

15. Tomasetti, C., Marchionni, L., Nowak, M. A., Parmigiani, G. & Vogelstein, B. Only three driver gene mutations are required for the development of lung and colorectal cancers. *Proc. Natl. Acad. Sci.* **112,** 118–123 (2015).

16. Nordling, C. O. A new theory on the cancer-inducing mechanism. *Br. J. Cancer* **7,** 68–72 (1953).

17. Armitage, P. & Doll, R. The age distribution of cancer and a multi-stage theory of carcinogenesis. *Br. J. Cancer* **8,** 1–12 (1954).

18. Hyman, D. M. *et al.* The efficacy of larotrectinib (LOXO-101), a selective tropomyosin receptor kinase (TRK) inhibitor, in adult and pediatric TRK fusion cancers. *J. Clin. Oncol.* **35,** LBA2501-LBA2501 (2017).

19. Svedberg, T. & Robin, F. A new method for the determination of the molecular weight of the proteins ACS. *J. Am. Chem. Soc.* **48,** 430–438 (1926).

20. Svedberg, T. Mass and size of protein molecules. *Nature* **123,** 871 (1929).

21. Jones, S. & Thornton, J. M. PROTEIN-PROTEIN INTERACTIONS: A REVIEW OF PROTEIN DIMER STRUCTURES SUSAN. *Prog. Biophys. Mol. Biol.* **63,** 31–65 (1995).

22. Hardy, S. J. S., Holmgrent, J. A. N., Johanssontt, S., Sanchezt, J. & Hirst, T. R. Coordinated assembly of multisubunit proteins : Oligomerization of bacterial enterotoxins in vivo and in vitro. *Proc. Natl. Acad. Sci.* **85,** 7109–7113 (1988).

23. Fields, S. & Song, O. A novel genetic system to detect protein protein interactions. *Nature* **340,** 245 (1989).

24. Auerbach, D. & Stagljar, I. Yeast Two-Hybrid Protein–Protein Interaction Networks. in *Proteomics and Protein-Protein Interactions* 19–32 (2005).

25. Phizicky, E. M. & Fields, S. Protein-Protein Interactions : Methods for Detection and Analysis. *Microbiol. Rev.* **59,** 94–123 (1995).

26. Macbeath, G. & Schreiber, S. L. Printing Proteins as Microarrays for High-Throughput Function Determination. *Science (80-.).* **289,** 1760–1764 (2000).

27. Evolutionary, I. F. Prediction of protein–protein interactions from evolutionary information. in *Structural Bioinformatics, Second Edition* 617–634 (2009).

28. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D. & Yeates, T. O. Assigning protein functions by comparative genome analysis : Protein phylogenetic profiles. *Proc. Natl. Acad. Sci.* **96,** 4285–4288 (1999).

29. Pazos, F. & Valencia, A. Similarity of phylogenetic trees as indicator of protein–protein interaction. *Protein Eng.* **14,** 609–614 (2001).

30. Enright, A. J., Iliopoulos, I., Kyrpides, N. C. & Ouzounis, C. A. Protein interaction maps for complete genomes based on gene fusion events. *Nature* **402,** (1999).

31. Thomas, J., Milward, D., Ouzounis, C., Pulman, S. & Carroll, M. Automatic extraction of protein interactions from scientific abstracts. in *Biocomputing 2000* 541–552 (World Scientific, 1999).

32. Andrade, A. automatic extraction of biological information from scientific text. *Ismb* **7,** 60–67 (1999).

33. Marcotte, E. M., Xenarios, I. & Eisenberg, D. Mining literature for protein–protein interactions. *Bioinformatics* **17,** 359–363 (2001).

34. Stuart, J. M., Segal, E., Koller, D. & Kim, S. K. A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science (80-.).* **302,** 249–256 (2003).

35. Mering, C. Von, Huynen, M., Jaeggi, D. & Schmidt, S. STRING : a database of predicted functional associations between proteins. *Nucleic Acids Res.* **31,** 258–261 (2003).

36. Wu, G., Feng, X. & Stein, L. A human functional protein interaction network and its application to cancer data analysis. *Genome Biol.* **11,** (2010).

37. Schadt, E. E. Molecular networks as sensors and drivers of common human diseases. *Nature* **461,** (2009).

38. Downward, J. TARGETING RAS SIGNALLING PATHWAYS IN CANCER THERAPY. *Nat. Rev. Cancer* **3,** (2003).

39. Shields, J. M., Pruitt, K., Shaub, A. & Der, C. J. Understanding Ras : ' it ain ' t over ' til it ' s over '. *Trends Cell Biol.* **8924,** 147–154 (2000).

40. Ouyang, L. *et al.* Programmed cell death pathways in cancer : a review of apoptosis , autophagy and programmed necrosis. *Cell Prolif.* **45,** 487–498 (2012).

41. Ogata, H. *et al.* KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **27,** 29–34 (1999).

42. Kanehisa, M. MAPK Signaling Pathway. Available at: https://www.genome.jp/kegg-bin/show_pathway?hsa04010.

43. Gillespie, M., Vastrik, I., Eustachio, P. D., Schmidt, E. & Bono, B. De. Reactome : a knowledgebase of biological pathways. *Nucleic Acids Res.* **33,** 428–432 (2005).

44. Kanehisa, M. A database for post-genome analysis. *Trends Genet.* **13,** 375–376 (1997).

45. Ferna, M. & Galperin, M. Y. The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection Death Domain database. *Nucleic Acids Res.* **40,** 1–8 (2012).

46. Schaefer, C. F. *et al.* PID: The pathway interaction database. *Nucleic Acids Res.* **37,** 674–679 (2009).

47. Complexity, T. H. E., Computing, O. F. & Minimal, S. The complexity of computing steiner minimal. *SIAM* **32,** 835–859 (1977).

48. Byrka, J., Grandoni, F., Rothvoß, T. & Sanità, L. An improved LP-based approximation for Steiner tree. in *Proceedings of the forty-second ACM symposium on Theory of computing* 583–592 (ACM, 2010).

49. Archer, A., Bateni, M., Hajiaghayi, M. & Karloff, H. Improved Approximation Algorithms for Prize-Collecting Steiner tree and tsp. *SIAM* **40,** 309–332 (2011).

50. Huang, S. C. & Fraenkel, E. Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Sci. Signal.* **2,** ra40-ra40 (2009).

51. Ljubić, I. *et al.* An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Math. Program.* **105,** 427–449 (2006).

52. Bailly-Bechet, M. *et al.* Finding undetected protein associations in cell signaling by belief propagation. *Proc. Natl. Acad. Sci. U. S. A.* **108,** 882–7 (2011).

53. Tuncbag, N. *et al.* Simultaneous reconstruction of multiple signaling pathways via the prize-collecting steiner forest problem. *J. Comput. Biol.* **20,** 124–136 (2013).

54. Gitter, A. *et al.* Sharing information to reconstruct patient specific pathways in heterogeneous diseases. in *Pacific Symposium on Biocomputing* **8,** 1385–1395 (2014).

55. Akhmedov, M. *et al.* PCSF : An R-package for network-based interpretation of high-throughput data. *PLoS Comput. Biol.* **13,** 1–7 (2017).

56. Akhmedov, M. *et al.* A fast prize-collecting steiner forest algorithm for functional analyses in biological networks. in *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* 263–276 (Springer, 2017).

57. Fischetti, M. *et al.* Thinning out Steiner trees : a node-based model for uniform edge costs. *Math. Program. Comput.* **9,** 203–229 (2017).

58. Dees, N. D. *et al.* MuSiC: Identifying mutational significance in cancer genomes. *Genome Res.* **22,** 1589–1598 (2012).

59. Lawrence, M. S. *et al.* Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature* **499,** 214–218 (2013).

60. Ciriello, G., Cerami, E., Sander, C. & Schultz, N. Mutual exclusivity analysis identifies oncogenic network modules. *Genome Res.* **22,** 398–406 (2012).

61. Leiserson, M. D. M. *et al.* Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.* **47,** 106–114 (2014).

62. Paull, E. O. *et al.* Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE). *Bioinformatics* **29,** 2757–2764 (2013).

63. Bashashati, A. *et al.* DriverNet: uncovering the impact of somatic driver mutations on transcriptional networks in cancer. *Genome Biol.* **13,** R124 (2012).

64. Cheng, F., Zhao, J. & Zhao, Z. Advances in computational approaches for prioritizing driver mutations and significantly mutated genes in cancer genomes. *Brief. Bioinform.* **17,** 642–656 (2016).

65. Tokheim, C. J., Papadopoulos, N., Kinzler, K. W., Vogelstein, B. & Karchin, R. Evaluating the evaluation of cancer driver genes. *Proc. Natl. Acad. Sci.* **113,** 14330–14335 (2016).

66. Beroukhim, R. *et al.* Assessing the significance of chromosomal aberrations in cancer : Methodology and application to glioma. *PNAS* **104,** 20007–20012 (2007).

67. Taylor, B. S. *et al.* Functional Copy-Number Alterations in Cancer. *PLoS One* **3,** (2008).

68. Milo, R., Kashtan, N., Itzkovitz, S., Newman, M. E. J. & Alon, U. Uniform generation of random graphs with arbitrary degree sequences. submitted to Phys. *Rev. E* (2001).

69. Garraway, L. A. & Lander, E. S. Lessons from the cancer genome. *Cell* **153,** 17–37 (2013).

70. Hou, J. P. & Ma, J. DawnRank: Discovering personalized driver genes in cancer. *Genome Med.* **6,** 1–16 (2014).

71. Guo, W. F. *et al.* Discovering personalized driver mutation profiles of single samples in cancer by network control strategy. *Bioinformatics* **34,** 1893–1903 (2018).

72. Shrestha, R. *et al.* HIT ' nDRIVE : Patient-Specific Multi-Driver Gene Prioritization for Precision Oncology. *Genome Res.* **27,** 1573–1588 (2017).

73. Vaske, C. J. *et al.* Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. *Bioinformatics* **26,** 237–245 (2010).

74. Page, L., Brin, S., Motwani, R. & Winograd, T. *The PageRank citation ranking: Bringing order to the web.* (Stanford InfoLab, 1999).

75. Wang, X., Tao, T., Sun, J.-T., Shakery, A. & Zhai, C. Dirichletrank: Solving the zero-one gap problem of pagerank. *ACM Trans. Inf. Syst.* **26,** 10 (2008).

76. Young, H. P. Condorcet's theory of voting. *Am. Polit. Sci. Rev.* **82,** 1231–1244 (1988).

77. Rasmussen, C. E. Factor Graphs and message passing. Available at: http://mlg.eng.cam.ac.uk/teaching/4f13/1718/factor graphs.pdf.

78. Pearl, J., Science, A. & Angeles, L. Reverand bayes on inference engines. in *AAAI* 133–136 (1982).

79. Barabási, A.-L. *Linked: The new science of networks*. (AAPT, 2003).

80. Bavelas, A. A mathematical model for group structures. *Appl. Anthropol.* **7,** 16–30 (1948).

81. Leavitt, H. J. Some effects of certain communication patterns on group performance. *J. Abnorm. Soc. Psychol.* **46,** 38 (1951).

82. Burgess, R. L. Communication Networks and Behavioral Consequences! *Hum. Relations* **22,** 137–159 (1969).

83. Freeman, L. C. Centrality in Social Networks Conceptual Clarification. *Soc. Networks* **1,** 215–239 (1978).

84. Jeong, H., Mason, S. P., Barabási, A.-L. & Oltvai, Z. N. Lethality and centrality in protein networks. *Nature* **411,** 41 (2001).

85. Bavelas, A. Communication Patterns in Task-Oriented Groups. *J. Acoust. Soc. Am.* **725,** (2015).

86. Szklarczyk, D. *et al.* STRING v10: Protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* **43,** D447–D452 (2015).

87. Joshi-Tope, G. *et al.* Reactome: A knowledgebase of biological pathways. *Nucleic Acids Res.* **33,** 428–432 (2005).

88. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15,** 1–21 (2014).

89. McLachlen, G. & Peel, D. *Finite Mixture Models*. (Wiley inter-science, 2000).

90. Forbes, S. A. *et al.* COSMIC: Somatic cancer genetics at high-resolution. *Nucleic Acids Res.* **45,** D777–D783 (2017).

91. Koboldt, D. C. *et al.* Comprehensive molecular portraits of human breast tumours. *Nature* **490,** 61–70 (2012).

92. Weinstein, J. N. *et al.* Comprehensive molecular characterization of urothelial bladder carcinoma. *Nature* **507,** 315–322 (2014).

93. Muzny, D. M. *et al.* Comprehensive molecular characterization of human colon and rectal cancer. *Nature* **487,** 330–337 (2012).

94. Lawrence, M. S. *et al.* Comprehensive genomic characterization of head and neck squamous cell carcinomas. *Nature* **517,** 576–582 (2015).

95. Collisson, E. A. *et al.* Comprehensive molecular profiling of lung adenocarcinoma: The cancer genome atlas research network. *Nature* **511,** 543–550 (2014).

96. Jonsson, P. F. & Bates, P. A. Global topological features of cancer proteins in the human interactome. *Bioinformatics* **22,** 2291–2297 (2006).

97. Porta-Pardo, E., Garcia-Alonso, L., Hrabe, T., Dopazo, J. & Godzik, A. A Pan-Cancer Catalogue of Cancer Driver Protein Interaction Interfaces. *PLoS Comput. Biol.* **11,** 1–18 (2015).

98. Dorsam, R. T. & Gutkind, J. S. G-protein-coupled receptors and cancer. *Nat. Rev. Cancer* **7,** 79–94 (2007).

99. Cotto, K. C. *et al.* DGIdb 3.0: a redesign and expansion of the drug–gene interaction database. *Nucleic Acids Res.* **46,** 1068–1073 (2017).

100. Van Allen, E. M. *et al.* Whole-exome sequencing and clinical interpretation of formalin-fixed, paraffin-embedded tumor samples to guide precision cancer medicine. *Nat. Med.* **20,** 682–688 (2014).

101. Kroschinsky, F. *et al.* New drugs , new toxicities : severe side effects of modern targeted and immunotherapy of cancer and their management. *Crit. Care* **21,** 1–11 (2017).

102. Park, S. R., Davis, M., Doroshow, J. H. & Kummar, S. Safety and feasibility of targeted agent combinations in solid tumours. *Nat. Rev. Clin. Oncol.* **10,** 154–168 (2013).

103. Hart, G. T., Ramani, A. K. & Marcotte, E. M. How complete are current yeast and human protein-interaction networks ? *Genome Biol.* **7,** (2006).

104. Sprinzak, E., Sattath, S. & Margalit, H. How Reliable are Experimental Protein – Protein Interaction Data ? *J. Mol. Biol.* **2836,** 919–923 (2003).

105. Ulitsky, I. & Shamir, R. Identification of functional modules using network topology and high-throughput data. *BMC Syst. Biol.* **17,** 1–17 (2007).

106. Silverbush, D. *et al.* ModulOmics: Integrating Multi-Omics Data to Identify Cancer Driver Modules. *bioRxiv* (2018).

107. Basha, O. *et al.* MyProteinNet: build up-to-date protein interaction networks for organisms, tissues and user-defined contexts. *Nucleic Acids Res.* **43,** W258–W263 (2015).

7. Supplementary material

Supplementary methods

Precision, recall, F1: Let $rank_i[k]$ be the set of top k ranking genes for the patient i, then

$$precision(rank_i[k]) = \frac{|rank_i[k] \cap CGC\_genes|}{k}$$

$$Recall(rank_i[k]) = \frac{|rank_i[k] \cap CGC\_genes|}{|SNV_i \cap CGC\_genes|}$$

where SNV_i is the set of mutated genes in patient i

$$F1(rank_i[k]) = 2 * \frac{\text{Precision } (rank_i[k]) \ * \ \text{Recall } (rank_i[k])}{\text{Precision } (rank_i[k]) \ + \ \text{Recall } (rank_i[k])}$$

In order to calculate the quality of prediction for a cohort of patients, we averaged these quality measures over the entire cohort and showed the results as a function of k.

Driver-Pathway linkage identification pipeline

1) First, we extract all driver-deregulated pathway pairs (g, pw) that are observed together in more than three patients. Here we defined the set of drivers for each patient as the 10 top ranked genes.

2) For each driver-pathway pair (g, pw) let n_g be the number of patients for whom the gene g was predicted as driver by Prodigy, n_{pw} the number of patients for whom the pathway pw was deregulated and $n_{g,pw}$ the number of patients for whom g was predicted as a driver and pw was deregulated. Let N be the number of patients in the cohort. The probability of observing $n_{g,pw}$ patients is given by $hypergeometric(N, n_g, n_{pw}, n_{g,pw})$. We calculate the p-value for $n_{g,pw}$ of each pair and identify the *significant pairs* (FDR < 0.05).

3) For each significant pair (g_i, pw_i), we perform a comparison between the absolute deregulation of pw_i in patients for whom g_i was predicted to be a driver and in patients for whom g_i was not predicted to be a dirver using t-test (**SFig 3**). All pairs with statistically significant difference (FDR < 0.05) were classified as *significant interactions.*

4) To ensure that these interactions could not have been identified by observing the mutational state of the gene alone (i.e., regardless of its ranking), we ran the entire process on all mutated genes g to identify *mutation-pathway* interactions. We then excluded significant driver-pathway interactions that overlapped significant mutation-pathway interactions. The driver-pathway pairs that passed this filter are reported as output.

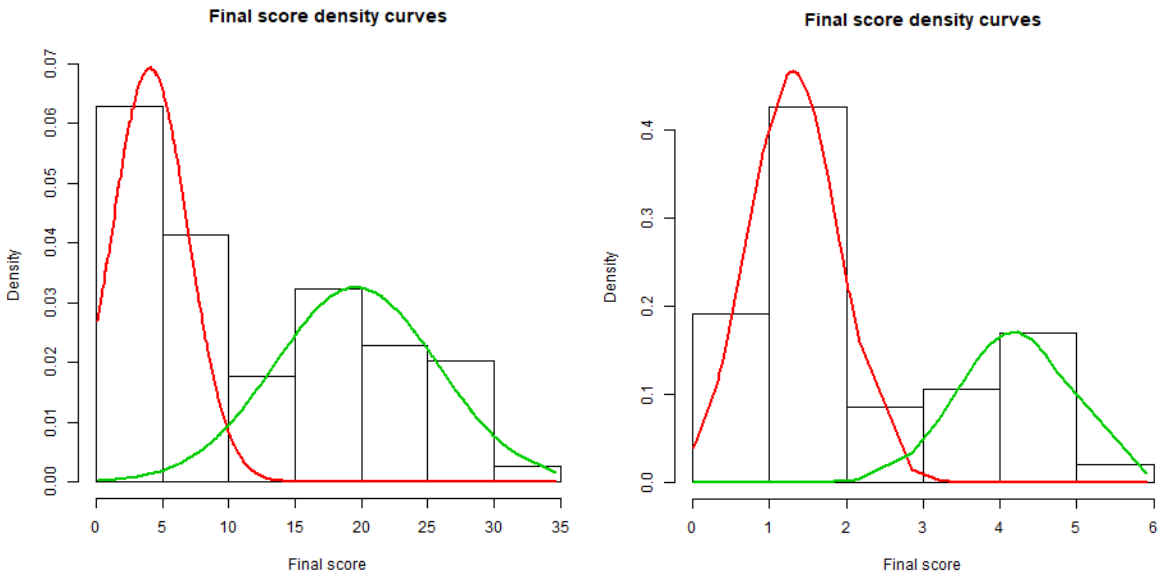
Multiple-pathways effect: We repeated the following procedure for each value of *k* between 1 and 50. For each gene g we used only the *k* pathways with the highest scores to calculate the gene's influence score $infl(g)$, and ranked all genes according to the score. We then computed the AUPR for the ranked list.

In order to examine the extent of overlap between the top *k* scoring pathways for each gene g_i, we computed for every pathway pw_i in the top *k* pathways its maximum overlap with any other pathway among the top *k*, and averaged these scores over all the pathways as follows:

$$Overlap\ Index(g_i) = \frac{1}{k}\sum_{i=1}^{k} \max_{pw_i \neq pw_j} JI(pw_i, pw_j).$$

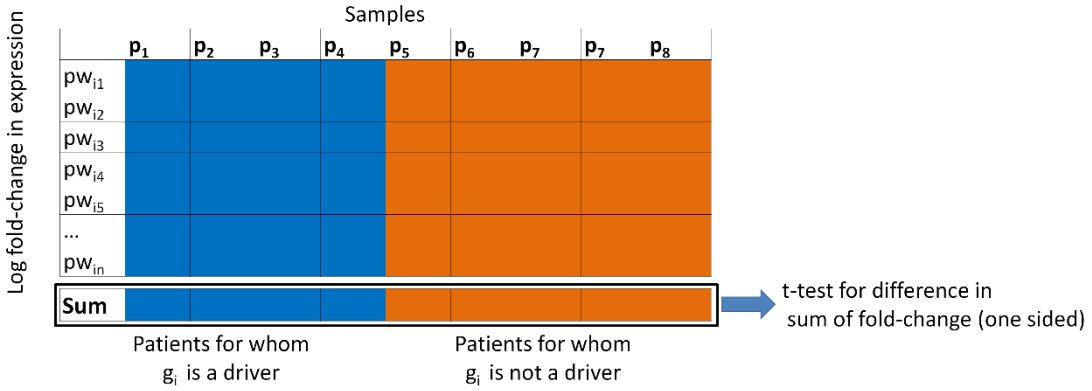
Here JI indicates the Jaccard Index, namely the number of genes shared by the two pathways divided by the number of genes in their union. We then averaged these values over all drivers to get the *Redundancy score*, a value that represents the extent of pathway overlap for the patient.

To explore the effect of this redundancy on the performance of Prodigy, we built the set of top pathways from which we derive the ranking in a sequential manner while avoiding redundancy: for each gene g we ranked the pathways by their scores $infl(p, g)$ in decreasing order and added the next pathway to the set only if it had JI < θ with every pathway already in the set. We did so for all the genes mutated in the patient, ranked the genes according to their final influence score and calculated AUPR for the patient. The results in **Figure 5C** show the mean AUPR across all patients as a function of θ and *k.*

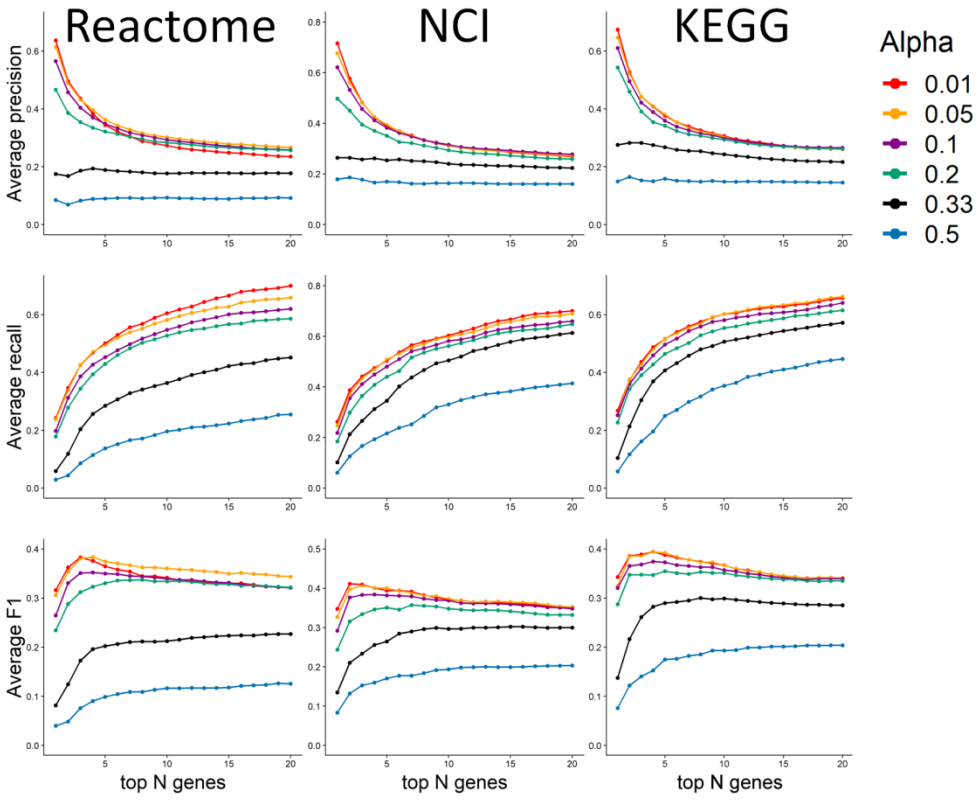


Supplementary Figure 1: Distribution of gene influence scores. The color lines show breakdown of the distribution into two Gaussians as detected by the EM algorithm. The red distribution reflects mutations that gain sporadic or low scores for few pathways due to the topology of the network. The green

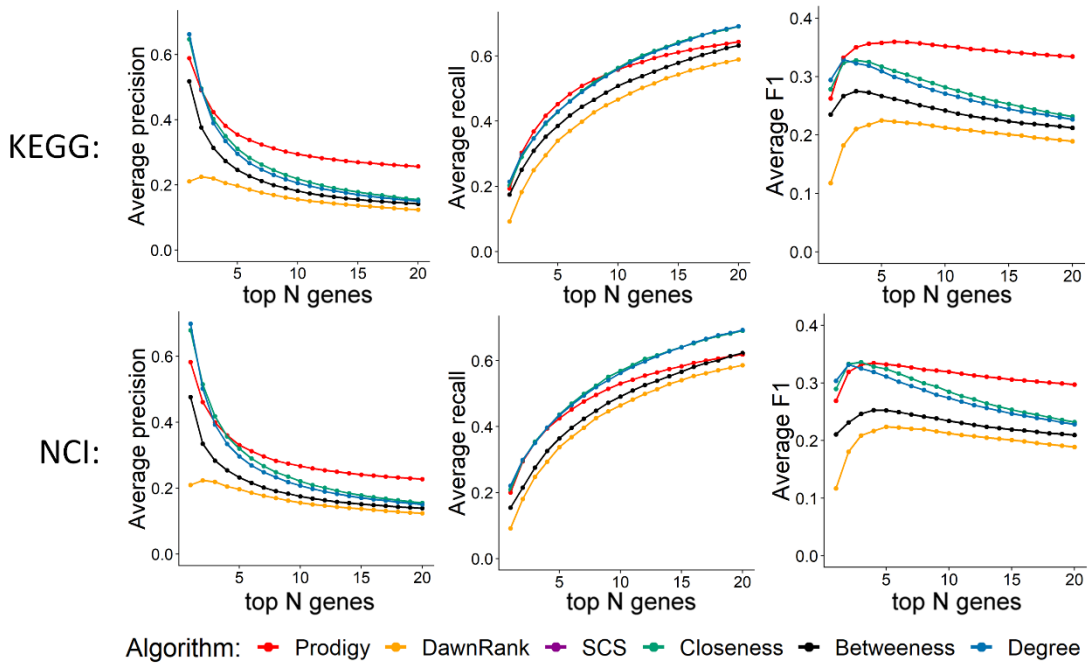
distribution reflects mutations with consistent scores across many pathways.



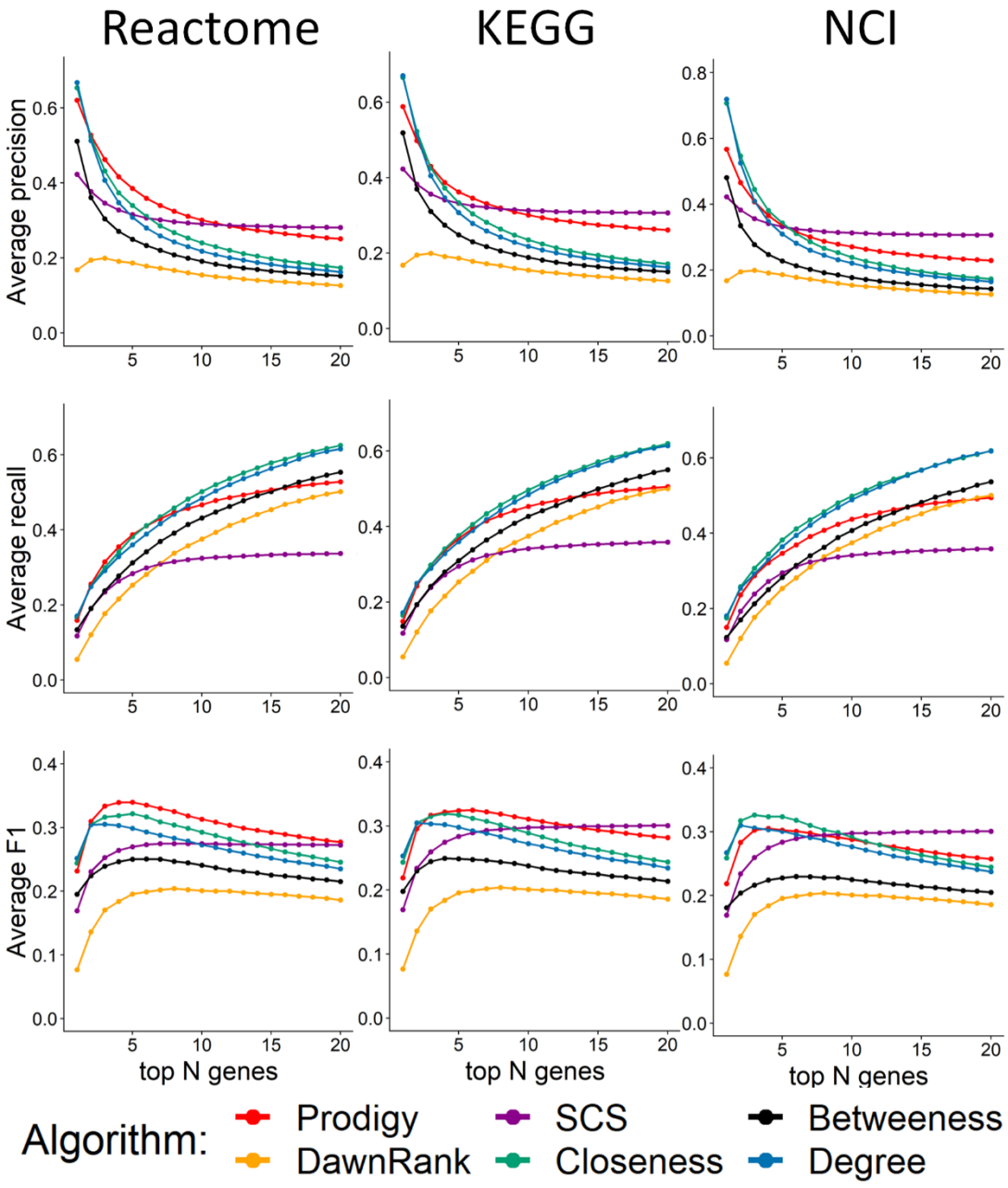
Supplementary Figure 2: Let (g_i, pw_i) be a significant pair. The matrix contains the log2 fold-change in the expression of genes that belong to pw_i (the genes $\{pw_{i1}, \dots, pw_{in}\}$). Columns in the blue / orange part of the matrix represent patients for whom g_i was / was not identified as a driver by Prodigy. A t-test is performed based on the total sum of fold-changes for each patient. The null hypothesis is that the mean sum of fold-changes is not greater in the group of patients for whom g_i was classified as a driver.



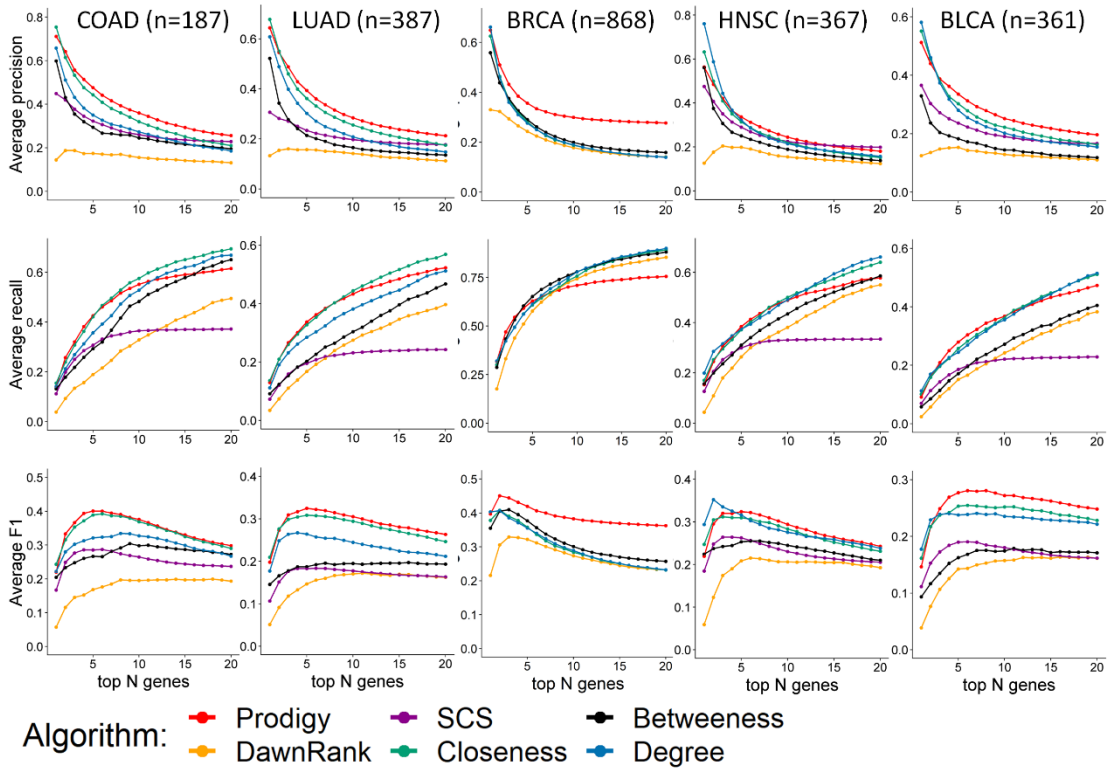
Supplementary Figure 3: Effect of the parameter α on Prodigy's results. The plots show average precision, recall and F1 on the training cohort (n=215) as a function of α, the exponent of the penalty term for Steiner nodes and N, the number of top ranked genes. The training cohort was comprised of 10% from each of the five datasets used.



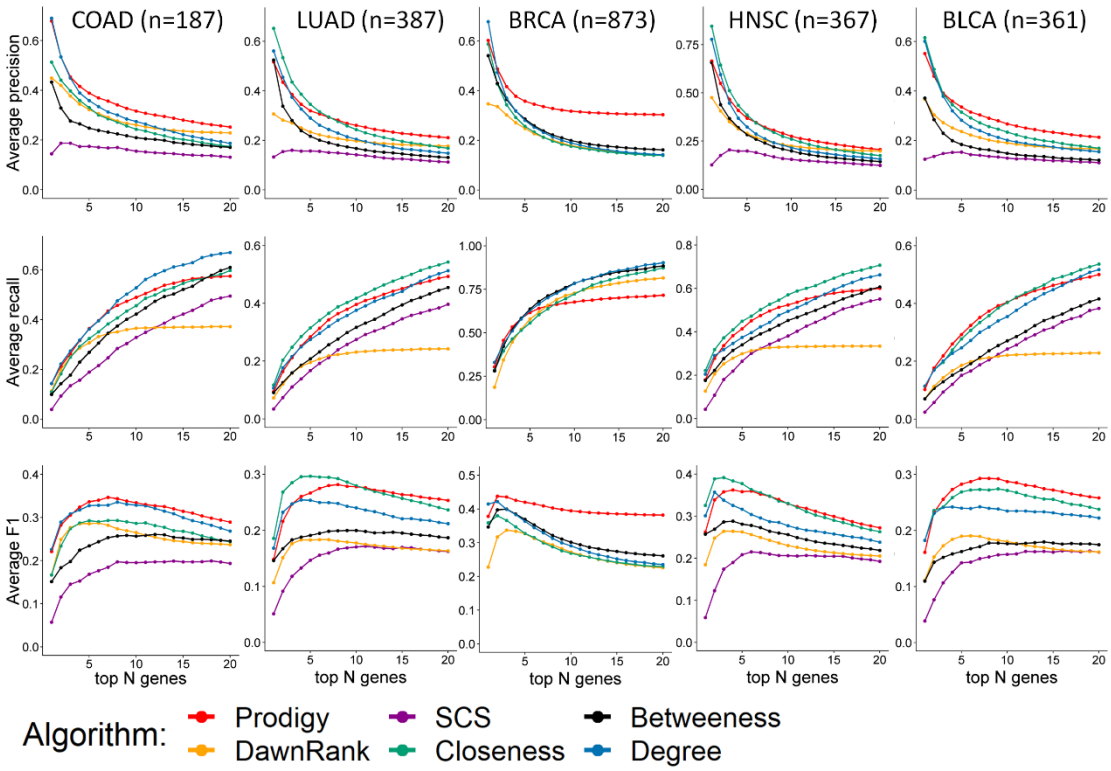
Supplementary Figure 4: Average precision, recall and F1 across all patients as a function of the top N genes in the personalized profiles. Results are for KEGG and NCI as pathway databases. Prodigy produced empty rankings for 14, 6, and 6 samples (<0.5%) in Reactome, KEGG and NCI, respectfully, due to lack of deregulated pathways or zero influence scores for all genes.



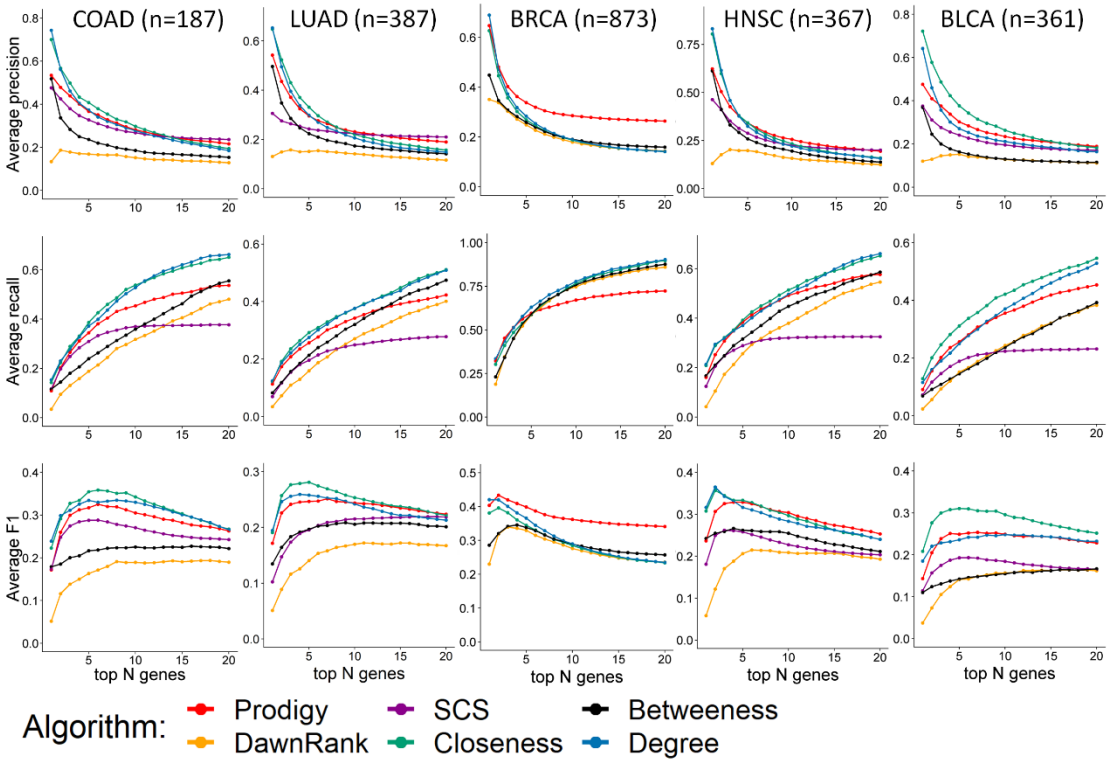
Supplementary Figure 5: Average precision, recall and F1 across all cohorts are presented as a function of the top N genes in the personalized profiles. The results are for the subset of samples for which SCS produced non-empty profiles (n=1847, 1849, 1849 for Reactome, KEGG and NCI).



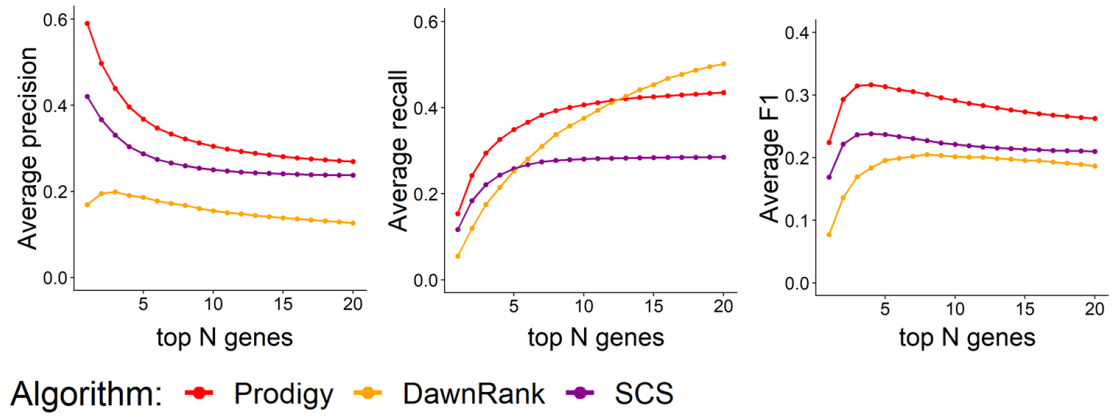
Supplementary Figure 6: Performance of the methods on each cohort using Reactome pathways. Results are shown for Prodigy, DawnRank, SCS and three centrality measures. The results of Prodigy and of the centrality measures were derived using STRING as global network and Reactome as pathway DB. SCS and DawnRank used their directed network. Average precision, recall and F1 across the relevant cohort are presented as a function of the top N genes. Results are for the "SCS sub cohort" (see **Methods**), except for the BRCA cohort where SCS produced empty rankings for 529 (55%) patients, hence it was excluded from the comparison. $\alpha = 0.05$ was used for all cohorts. n = sample size of the test group.



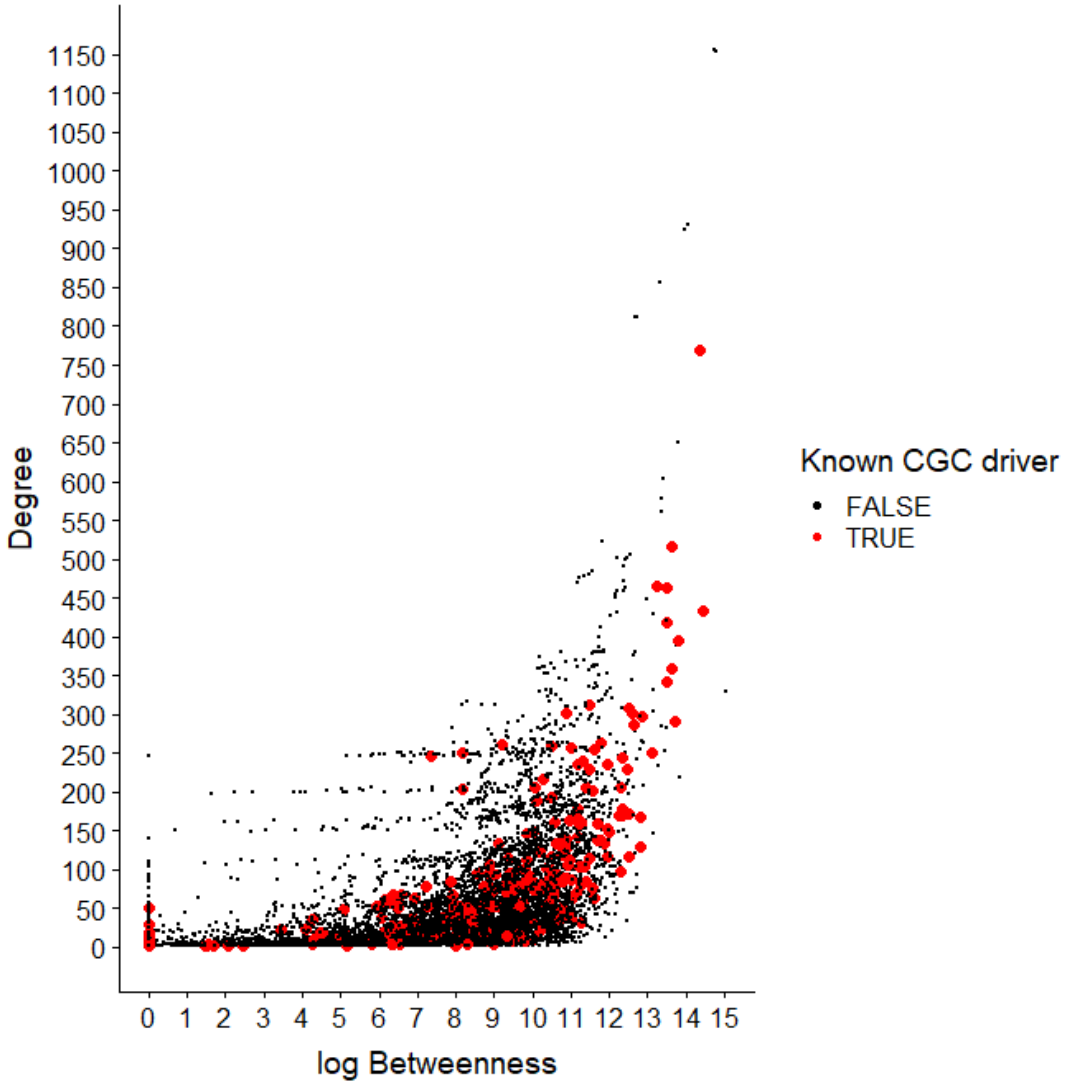
Supplementary Figure 7: Performance of the methods on each cohort using KEGG pathways. Details are as in Supplementary Figure 7 but here the KEGG pathway DB was used.



Supplementary Figure 8: Performance of the methods on each cohort using NCI pathways. Details are as in Supplementary Figure 7 but here the NCI pathway DB was used.



Supplementary Figure 9: Average precision, recall and F1 based on the adjusted underlying network from[1,2] (see **Methods**). Results of Prodigy are based on Reactome as pathway DB and for $\alpha = 0.05$. Cohort size: 1804.



Supplementary Figure 10: Centrality measures for all nodes in the global STRING network used in this study. Each point represent a different gene in the network (n = 11,302). The position on the X axis is the log betweenness of the gene and the Y axis is its degree. Known drivers (from CGC) are colored red. Known drivers tend to have higher degrees in the network and higher betweenness values (Wilcoxon rank sum test p-value < 2.2^-16 for both).

Cohort / # mutations per mutated gene (%)	1	2	3	>3
COAD	66817 (91.1)	5167 (7.04)	910 (1.24)	423 (0.57)
LUAD	108946 (82.81)	16218 (12.32)	1400 (1.06)	4990 (3.79)
BRCA	16074 (41.17)	18224 (46.68)	4126 (10.56)	614 (1.57)
HNSC	73194 (95.59)	2984 (3.89)	263 (0.34)	122 (0.15)
BLCA	86066 (94.52)	4209 (4.62)	553 (0.6)	221 (0.24)

Supplementary Table 1: Number of non-silent mutations per mutated gene in different cohorts.

Pathway database	Reactome	KEGG	NCI
Number of pathways used	1762	285	212
Mean number of interactions per pathway (SD)	1019.6 (3078.9)	417.6 (849)	109.6 (107.53)
Mean number of genes per pathway (SD)	58.7 (143.31)	65.2 (59)	33.5 (20.2)

Supplementary Table 2: Pathway database statistics.

Cohort	COAD	BRCA	LUAD	HNSC	BLCA
Number of observed rarely mutated genes (%)	24,144 (36.1)	55,279 (93.8)	26,682 (42.6)	44,831 (70.1)	43,248 (53.9)
Number of rare mutations in genes ranked by Prodigy (%)	403 (21)	4405 (78.4)	1610 (37.1)	2026 (52.9)	1364 (38.6)

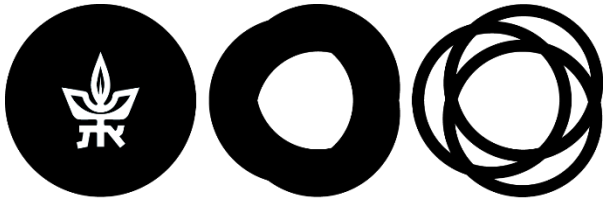
Supplementary Table 3: The number of overall and ranked rarely mutated genes. A gene is rarely mutated in a cohort if it has at least one mutation in < 2% of the patients. For example, since 93.8% of all observed mutated genes in BRCA were mutated in < 2% of patients, the overwhelming majority of mutated genes in BRCA patients are rare. The second row describes the number of rarely mutated genes that were prioritized in the top 10 rankings by Prodigy.

תקציר

רקע: סרטן הוא תהליך אבולוציוני המונע ממספר קטן של מוטציות בתאים סומטיים הנקראות "מוטציות נהג" (driver mutations) והגנים בהם המוטציות מתרחשות נקראים "גני נהג" (driver genes). מוטציות אלו משבשות את מנגנוני התא הטבעיים וגורמות לחלוקת תא בלתי נשלטת המתפתחת לגידול, בעוד שרוב המוטציות הסומטיות הקיימות בתאים סרטניים אינן משפיעות על התקדמות המחלה. מוטציות אלו נקראות "מוטציות נוסע" (passenger mutations). זיהוי המוטציות המניעות את התהליך הסרטני מבין כלל המוטציות הקיימות בגידול בגוף החולה הינו אחד היסודות לתוויית טיפול אישי: מידע על מוטציות הנהג והמנגנונים הביולוגיים עליהם הן משפיעות יכול לשפוך אור על דרכי טיפול אפשריים. עד כה, המחקר בתחום התמקד בעיקר בזיהוי מוטציות נהג בקרב אוכלוסיות חולים גדולות, אך זיהוי מוטציות אלו בצורה מותאמת אישית היא אתגר עליו שמה הקהילה האקדמית פחות דגש.

שיטות: בעבודה זו, פיתחנו אלגוריתם לתעדוף מותאם אישית של גני נהג. האלגוריתם, הנקרא PRODIGY, מנתח את פרופילי המוטציות וביטוי הגנים של החולה ומשתמש במידע אודות מסלולים ביולוגיים ידועים ורשת קשרי חלבון-חלבון גדולה. האלגוריתם מכמת את ההשפעה של כל מוטציה על כל מסלול ביולוגי בלתי מבוקר בעזרת מודל גרפי המבוסס על עצי שטיינר (Steiner strees). המוטציות מדורגות על פי ההשפעה הכוללת שלהן על כל המסלולים הבלתי מבוקרים.

תוצאות: מבדיקה שערכנו על יותר מ-2500 דגימות גידול שנלקחו מחמש אוכלוסיות חולים מה-TCGA והשוואה לגני נהג ידועים, עולה כי Prodigy הוא בעל ביצועים טובים יותר מהשיטות הקיימות ומשיטות המבוססות על מדדי מרכזיות רשתית. התוצאות שלנו מדגימות את ההשפעה רחבת ההיקף של גני הנהג על מסלולים ביולוגיים רבים ומראות כי Prodigy מסוגל לזהות אפילו מוטציות נדירות ביותר. Prodigy יכול לסייע לאונקולוגים בהתוויית טיפול מותאם אישי לחולים, על ידי התאמת הטיפול לדירוג המוטציות של החולה.



אוניברסיטת תל אביב

הפקולטה למדעים מדוייקים ע"ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב ע"ש בלווטניק

תעדוף מותאם אישית של גנים סרטניים

חיבור זה הוגש כעבודת גמר לתואר "מוסמך אוניברסיטה" בבית הספר

למדעי המחשב

על ידי

גל דינסטג

בהנחיית

פרופ' רון שמיר

כסלו תשע"ט