Bioinformatics, 36, 2020, i508–i515
doi: 10.1093/bioinformatics/btaa452
ISMB 2020

Prediction of cancer driver genes through network-based moment propagation of mutation scores

Anja C. Gumpinger[1,2,*], Kasper Lage[3,4], Heiko Horn[3,4,*] and Karsten Borgwardt[1,2,*]

[1]Department of Biosystems Science and Engineering, Machine Learning and Computational Biology Lab, ETH Zürich, Basel 4058, Switzerland, [2]SIB Swiss Institute of Bioinformatics, Lausanne 1015, Switzerland, [3]Department of Surgery, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, USA and [4]Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

*To whom correspondence should be addressed.

Abstract

Motivation: Gaining a comprehensive understanding of the genetics underlying cancer development and progression is a central goal of biomedical research. Its accomplishment promises key mechanistic, diagnostic and therapeutic insights. One major step in this direction is the identification of genes that drive the emergence of tumors upon mutation. Recent advances in the field of computational biology have shown the potential of combining genetic summary statistics that represent the mutational burden in genes with biological networks, such as protein–protein interaction networks, to identify cancer driver genes. Those approaches superimpose the summary statistics on the nodes in the network, followed by an unsupervised propagation of the node scores through the network. However, this unsupervised setting does not leverage any knowledge on well-established cancer genes, a potentially valuable resource to improve the identification of novel cancer drivers.

Results: We develop a novel node embedding that enables classification of cancer driver genes in a *supervised* setting. The embedding combines a representation of the mutation score distribution in a node's local neighborhood with network propagation. We leverage the knowledge of well-established cancer driver genes to define a positive class, resulting in a partially labeled dataset, and develop a cross-validation scheme to enable supervised prediction. The proposed node embedding followed by a supervised classification improves the predictive performance compared with baseline methods and yields a set of promising genes that constitute candidates for further biological validation.

Availability and implementation: Code available at https://github.com/BorgwardtLab/MoProEmbeddings.

Contact : anja.gumpinger@bsse.ethz.ch or karsten.borgwardt@bsse.ethz.ch or hhorn@broadinstitute.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Cancer is a disease of unchecked cellular growth, caused by genetic alterations such as mutations, copy number variations or gene fusions in so called *cancer driver genes*. Those alterations can modify both, the activity and cellular function of the gene, and can be classified into activating (proto-oncogenes), or loss of function (tumor suppressor genes and DNA repair genes). Identification of such cancer driver genes is one of the main goals of oncogenic research, as it facilitates mechanistic, diagnostic and therapeutic insights.

Cancer genes can be identified through statistical tests that evaluate the mutational burden of the gene (e.g. Kandoth *et al.*, 2013; Leiserson *et al.*, 2015; Mularoni *et al.*, 2016). However, those analyses are complicated by the extensive mutational heterogeneity: Many genes are mutated in a small number of samples, and only few genes show significant mutation across many samples (Vogelstein *et al.*, 2013). This phenomenon convolutes the differentiation between genes that only carry passenger mutations, and rarely mutated cancer genes. A potential explanation of this diversity in candidate genes is that genes interact in various pathways (Hanahan and Weinberg, 2011) and protein complexes, and the cancerous potential of a cell is a consequence of the disruption of the pathway, but not necessarily the mutation of *one* specific gene within the pathway.

Recent research adopted this interaction-based view on cancer biology: the combination of biological networks and summary statistics that measure each gene's association to cancer helped the identification of novel cancer driver genes (e.g. Horn *et al.*, 2018; Leiserson *et al.*, 2015; Reyna *et al.*, 2018). In those networks, nodes correspond to genes and edges represent relationships between the adjacent genes. There exists a vast number of biological networks, that are derived from different sources and that cover different scales. Prominent examples are co-expression networks (Willsey *et al.*, 2013), co-dependency networks (e.g. AchillesNet; Li *et al.*, 2018), co-evolution networks (Niu *et al.*, 2017), metabolic pathways (Kanehisa *et al.*, 2017) or protein–protein interaction (PPI) networks (Lage *et al.*, 2007; Li *et al.*, 2017; Szklarczyk *et al.*, 2019). Especially, PPI networks constitute an interesting representation of gene interactions, as they commonly combine information from

different data sources, tissues, and molecular processes at different scales. However, those PPI networks are far from complete, and our knowledge of them is biased toward well-studied genes (Horn *et al.*, 2018). This phenomenon is referred to as *knowledge contamination*: well-studied (cancer) genes have a tendency to have more connections in the networks. The potentially large impact on the interpretation of network analyses has to be considered and accounted for, as it might confound results.

Methods that use networks as a representation of molecular relationships commonly start with superimposing scores on the nodes. These scores measure the marginal association of the gene to the disease of interest. A prominent choice to represent each gene's association to cancer is the MutSig *P*-value (Lawrence *et al.*, 2014): it is a meta-*P*-value describing whether there is a statistically significant difference in (i) the mutational burden, (ii) the clustering of mutations and (iii) the functional impact of mutations in a gene between healthy and cancer tissues. A plethora of methods has been developed to analyze such gene scores in combination with network information to identify altered subnetworks of genes within the original network. They can be broadly categorized into clustering methods, that aim to find modules of associated genes that cluster together in a network (e.g. Jia *et al.*, 2011; Rossin *et al.*, 2011) and methods that use network diffusion or network propagation (reviewed in Cowen *et al.*, 2017; Reyna *et al.*, 2018) to detect altered subnetworks. Both types of methods underlie the common paradigm that genes influencing the same phenotype interact within a network. Especially network propagation methods have shown success in identifying novel cancer driver genes (Hristov *et al.*, 2020; Leiserson *et al.*, 2015; Reyna *et al.*, 2018; Ruffalo *et al.*, 2015; Vandin *et al.*, 2011, 2012). However, network propagation methods exploit by construction the flow of information between genes along paths, and the longer the paths are, the more information gets diluted. This complicates the detection of cancer genes that do not lie on short paths between other cancer genes.

Another approach that has proven successful and does not leverage this assumption is *NetSig*. It identifies cancer genes based solely on the local neighborhood of genes in a network (Horn *et al.*, 2018). At its core lies the computation of an empirical *P*-value for each gene that describes the aggregation of genes with low MutSig *P*-values in the direct neighborhood. Due to knowledge contamination, the size of a gene's local neighborhood is affecting the NetSig statistic. To circumvent this, NetSig implements various permutation schemes that take the node degree into account, thereby correcting for this bias.

Although the aforementioned methods showed great success in many biomedical applications (Cowen *et al.*, 2017), including the discovery of novel cancer genes, they approach the task of gene identification from an unsupervised perspective. However, there exists knowledge on well-established cancer genes (e.g Sondka *et al.*, 2018), an important layer of additional information that has, to the best of our knowledge, only been leveraged in few methods for the prediction of cancer driver genes, namely Bayesian modeling (Sanchez-Garcia *et al.*, 2014) and unsupervised network propagation (Hristov *et al.*, 2020). In most cases, well-established cancer genes are only used to validate the importance and correctness of findings from new methods as a post-processing step. It seems to be an interesting approach to reformulate the task of identifying novel cancer genes as a supervised problem, and learning by exploiting *what we already know*.

Herein, we propose a novel approach to classify cancer genes in a supervised manner, leveraging the cancer gene annotations from the Cancer Gene Census (CGC) in the COSMIC database (Sondka *et al.*, 2018). We achieve this by formulating the problem of finding novel cancer driver genes as a node-classification problem in an interaction network. The heart of our contribution is a novel embedding of nodes in the network based on the distributions of node-features in k-hop neighborhoods, coupled with a network propagation. We combine the InWeb PPI network (Lage *et al.*, 2007; Li *et al.*, 2017) with MutSig *P*-values (Lawrence *et al.*, 2014), and the CGC genes, resulting in an imbalanced dataset due to the low number of known cancer genes compared with the gene corpus. To address this, we develop a cross-validation scheme that enables the supervised prediction of cancer driver genes with a set of classifiers. We compare our approach against both, supervised and unsupervised baselines, and show an improvement with respect to all classification metrics. Last, we evaluate the resulting set of high confidence novel cancer driver candidate genes and find strong links between the predictions and cancer. The list includes known tumor suppressors such as GATA4 (Agnihotri *et al.*, 2011), genes known to be affected by recurrent rearrangements FOS (Fittall *et al.*, 2018) as well as genes known to be involved in tumor relevant pathways ID2 (Kijewska *et al.*, 2019), MYLK (Avizienyte *et al.*, 2005; Cui *et al.*, 2010; Zhou *et al.*, 2008), RALA (Seibold *et al.*, 2019).

2 Materials and methods

Before we present our novel node embedding procedure, we start by introducing the notation used in this Section, and formally state the problem at hand.

2.1 Notation and problem statement

Consider a PPI network that describes interactions between genes. We can represent this interaction network as a graph $\mathcal{G}$, where the n nodes correspond to the genes, and the m edges correspond to interactions between genes. We denote the vertex-set as V, and the edge set as E. We denote an edge between two nodes $u, v \in V$ as $e(u, v)$, and assume a weighting function $\omega : V \times V \to [0, 1]$ that assigns each pair of nodes in the network a value between 0 and 1, such that $\omega(u, v) > 0 \iff e(u, v) \in E$. In the case of a weighted network, the function ω might correspond to confidence scores of edges, in the case on an unweighted network, ω is a binary indicator. Additionally, we assume the existence of a d-dimensional feature representation for every vertex $v \in V$, denoted by $x_v \in \mathbb{R}^d$ or in matrix notation by $X \in \mathbb{R}^{n \times d}$. We write the graph as $\mathcal{G} = (V, E, X, \omega)$.

We define the k-hop neighborhood of a vertex $v \in V$ as the set of all genes that can be reached from v along at least k edges. This can be expressed recursively as

$$\mathcal{N}_v^k = \{u \,|\, e(w, u) \in E, \forall w \in \mathcal{N}_v^{k-1}, \; u \notin \mathcal{N}_v^l \, \forall l \in \{0, .., k-1\}\}, \tag{1}$$

where $\mathcal{N}_v^0 = v$ (see Fig. 1a for visualization of one- and two-hop neighborhoods).

Problem statement. We assume a partially labeled, one-class setting in which we have a positive label l_1 for a subset V_l of the nodes, but the majority of the nodes are unlabeled, denoted by the set V_u. Our goal is to develop a node embedding $\gamma_{\mathcal{G}}(\cdot)$ based on the feature representation X of all vertices $v \in V$ and the network $\mathcal{G}$ that, in combination with a binary classifier C, enables the decision of whether any unlabeled node $u \in V_u$ belongs to the class l_1. That is $C(\gamma_{\mathcal{G}}(x_u)) = \mathcal{P}(y_u = l_1)$, where y_v denotes the label of node v.

More specifically, our goal is to develop a node embedding that serves as input for the supervised, binary classification task of identifying cancer driver genes. It is based on the integration of a PPI network with scores that measure the marginal association of each gene to cancer, when those scores are superimposed on the nodes in the network.

2.2 Generation of node embeddings for the prediction of cancer driver genes

The node embeddings proposed in this article are based on two different concepts. The first one is the representation of each node as a distribution across its neighbors' feature vectors. It is motivated by the success of methods such as NetSig (Horn *et al.*, 2018) that focus on the local neighborhood of nodes in the network. However, we extend this idea in three directions: (i) we do not restrict ourselves to one-hop neighborhoods, (ii) we condense the distributions into their *moments*, resulting in a concise and computationally efficient representation and (iii) we integrate edge weights into the approach. This moment representation addresses the knowledge-bias in the

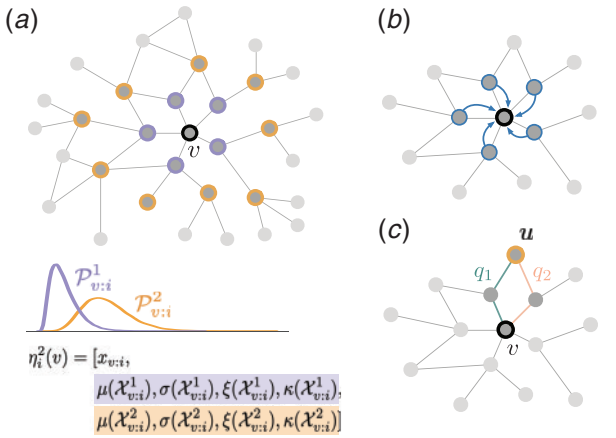

Fig. 1. Illustration of moment propagation embeddings for node feature i. (a) Computation of moment embedding $\eta_i^k(v)$ for vertex v and $k=2$. Blue nodes indicate the one-hop, orange nodes the two-hop neighborhood. Moments of the distributions $\mathcal{P}_{v:i}^1$ and $\mathcal{P}_{v:i}^2$ that describe the values of feature i in the one- and two-hop neighborhood of vertex v are computed and aggregated. (**b**) Computation of propagation embedding: The node representation is updated by aggregating over all nodes in its one-hop neighborhood. (**c**) Two paths q_1 and q_2 connect root vertex v with its two-hop neighbor u

network, as the description of a distribution via its moments is independent of the number of draws from that distribution. The second concept is a Weisfeiler–Lehman like (Weisfeiler and Lehmann, 1968; Shervashidze *et al.*, 2011) aggregation of local features, that is an iterative combination of features from a node's local neighborhood. It is similar to the network propagation approaches described in Cowen *et al.* (2017), and is widely used in methods such as hierarchical HotNet (Reyna *et al.*, 2018) and its earlier versions (Leiserson *et al.*, 2015; Vandin *et al.*, 2012), but also in graph convolutional networks (e.g. Gilmer *et al.*, 2017; Kipf and Welling, 2016), where different regimes for aggregation over local neighborhoods are being actively researched.

2.2.1 Embedding genes using moments of local neighborhood distributions

Each node $v \in V$ in the network is represented by its d-dimensional feature vector $x_v \in \mathbb{R}^d$, and we denote the ith feature by $x_{v:i}$. The moment embeddings described in this section are computed for each of the d features identically and independently, and eventually stacked together. We assume the existence of a probability distribution $\mathcal{P}_{v:i}^k$ that generates the ith feature for all nodes in the k-hop neighborhood of node v. That is, the ith feature values of k-hop neighbors of v constitute draws from this distribution, $\forall u \in \mathcal{N}_v^k : x_{u:i} \sim \mathcal{P}_{v:i}^k$. We create an embedding for every vertex v that is based on a concise description of the distributions $\mathcal{P}_{v:i}^k$ for $i = 1, .., d$, and hyperparameter $k \in \mathbb{N}_+$. For this, we start by defining a function $\bar{\nu}(X)$ that maps a scalar random variable $X \sim \mathcal{P}$ to its first four moments, that is:

$$\bar{\nu}(X) = [\mathbb{E}_X[X], \mathbb{E}_X[X^2], \mathbb{E}_X[X^3], \mathbb{E}_X[X^4])]. \tag{2}$$

In practice, the expectations in Equation (2) can be replaced with the sample mean $\mu(\cdot)$, variance $\sigma(\cdot)$, skewness $\xi(\cdot)$ and kurtosis $\kappa(\cdot)$ and applied to a realization of the random variable X, denoted by $\boldsymbol{x} = [x_1, \ldots, x_q]$. We write this function as

$$\nu(\boldsymbol{x}) = [\mu(\boldsymbol{x}),\ \sigma(\boldsymbol{x}),\ \xi(\boldsymbol{x}),\ \kappa(\boldsymbol{x})], \tag{3}$$

and call it a *moment embedding function*. For a vertex v, we denote with $\mathcal{X}_{v:i}^k$ the values of the ith feature of vertices in the k-hop neighborhood of v, i.e. $\mathcal{X}_{v:i}^k = \{x_{u:i} \,|\, u \in \mathcal{N}_v^k\}$. Those values constitute a draw from the distribution $\mathcal{P}_{v:i}^k$. We describe the node embedding of vertex v with respect to feature i by applying the function $\nu(\cdot)$ up to its k-hop neighborhoods, that is

$$\eta_i^k(v) = [x_{v:i}\,, \nu(\mathcal{X}_{v:i}^1), \ldots\,, \nu(\mathcal{X}_{v:i}^k)], \tag{4}$$

The value k is a hyperparameter of the embedding that indicates the maximum neighborhood to be included (see Fig. 1a for an example of the node embeddings). The moment embedding η_i^k is a function that creates a representation of every vertex $v \in V$ by describing its k-hop neighborhoods with respect to a scalar feature indexed by i, such that $\eta_i^k(v) \in \mathbb{R}^{(1+4k)}$. This function can be applied to each of the d node features separately, and the resulting representations are stacked to give

$$\eta^k(v) = [\eta_1^k(v)\,, \ldots\,, \eta_d^k(v)]^T. \tag{5}$$

This results in the moment embedding function $\eta^k : V \to \mathbb{R}^{d\times(1+4k)}$.

2.2.2 Embeddings using network propagation

The second type of node embeddings is based on a Weisfeiler–Lehman like aggregation of nodes in the neighborhood with continuous node features. In this procedure, the representation of every node is simultaneously updated based on the representations of the node's direct neighborhood (see Fig. 1b for an example). That is, given an initial feature representation x_v of vertex v, it is represented as

$$x_v^t = \frac{1}{|\mathcal{N}_v^1|} \sum\nolimits_{v' \in \mathcal{N}_v^1} x_{v'}^{t-1} \tag{6}$$

at the tth Weisfeiler–Lehman iteration. This aggregation corresponds to the element-wise mean across a node's one-hop neighborhood, and can be used to generate node embeddings by stacking the representations for t iterations as follows:

$$\rho^t(v) = [x_v^0, x_v^1, \ldots, x_v^t], \tag{7}$$

and $\rho^t(v) : \mathbb{R}^d \to \mathbb{R}^{(1+t)\times d}$. The number t of iterations of this propagation scheme is treated as a hyperparameter that can be tuned during learning.

2.2.3 Combining moment and propagation embeddings to represent genes in a network

Here, we propose a combination of the two concepts introduced above, and call the resulting node embedding a *moment propagation embedding*, short *MoPro* embedding. It corresponds to a composition of the moment embeddings η^k and the propagation embedding ρ^t above, and can be written as:

$$\gamma_{t,k}(v) = (\rho^t \circ \eta^k)(v). \tag{8}$$

This function first creates the moment embedding from the feature vector x_v of a vertex v, and continues to propagate this representation of the local neighborhoods through the network. As the combination of both functions, it maps the original feature representations of the vertices to a higher dimensional space as follows: $\gamma_{t,k} : V \to \mathbb{R}^{(1+t)(1+4k)\times d}$.

2.2.4 Extension to networks with weighted edges

If a non-binary weighting function ω exists, i.e. the edges in the network are weighted and weights can for instance represent confidence scores, we can incorporate this layer of information into our approach: for every edge in the network, the value of the weighting function is non-zero, that is $\forall e(v,u) \in E : \omega(v,u) \in (0,1]$, with 1 indicating the highest confidence. These weights can be used to distribute importance of neighbors in a local neighborhood by rescaling the node-features in the moment embedding. This rescaling is done for each feature i separately, such that the values of features i in the k-hop neighborhood of node v become

$$\mathcal{X}_{v:i}^{k,\text{weight}} = \{f(x_{u:i}, \omega(v,u)) \,|\, x_u \in \mathcal{N}_v^k,\} \tag{9}$$

with a problem-specific weighting-function $f(\cdot,\cdot)$. For k-hop neighbors u of v with $k>1$ the weight $\omega(u,v)$ is zero by definition.

Hence, we compute it in the following three-step process, and denote it as $\omega^k(u,v)$: (i) First, we enumerate all paths of length k between v and u, denoted by the set $\mathcal{Q}^k$, and individual paths in $\mathcal{Q}^k$ are denoted as $q_j(v,u)$ (see Fig. 1c). (ii) Second, we compute the weight of each path in $\mathcal{Q}^k$ as the product of its k edge weights and (iii) third we compute the weight $\omega^k(v,u)$ as a function on the set of path weights, $g:[0,1]^{|\mathcal{Q}^k|}\to\mathbb{R}$. We treat this function $g(\cdot)$ as a hyperparameter, and use either $g(\cdot)=\max(\cdot)$ or $g(\cdot)=\text{mean}(\cdot)$.

3 Results

3.1 Dataset description

In order to find novel cancer driver genes, we combine data from a The Cancer Genome Atlas (TCGA) pan-cancer study of 9 423 tumor exomes (comprising all 33 of TCGA projects; Bailey *et al.*, 2018) with the well-established InBio Map PPI network (Lage *et al.*, 2007; Li *et al.*, 2017). The network constitutes our view of interactions between genes on a protein level. The network has an average degree of 61.02 ($\pm$128.33), and the sizes of the k-hop neighborhoods are illustrated in Figure 2a. We represent each node in the network with its $-\log_{10}$ transformed MutSig P-value (Lawrence *et al.*, 2014). Those P-values measure whether a gene shows significantly different mutational patterns in tumor versus normal tissues. In total, we have access to P-values for 18 154 genes. As a pre-processing step, we remove all nodes from the network that cannot be represented with a MutSig P-value, as well as all isolated nodes. This results in a total of 11 449 genes that are present in the InBio Map network, are connected to at least one other node and have been tested with the MutSig tool. Those constitute our candidates for network-based prediction of cancer driver genes.

Class labels. In general, supervised machine learning requires access to labeled data to train a classifier. To obtain labels for the genes, we use the CGC data from the COSMIC database (Sondka *et al.*, 2018). We downloaded a list of 723 genes that have been causally implicated in cancer, and use this set as our ground truth. Genes in the CGC are categorized into Tiers 1 and 2, where genes in Tier 1 show a documented activity relevant to cancer, and genes in Tier 2 show strong indications to play a role in cancer. For our analysis we treat both tiers equally. We overlap the set of 723 genes with our network, giving a total of 635 cancer genes. This leads to a dataset, in which 'positive' samples make up <6.0% of our dataset. We refer to the remaining genes as *unlabeled genes*, and we are interested in finding new cancer genes among them.

Using the CGC genes, we observe a knowledge-bias in the InBio Map PPI network (see Fig. 2b), that is cancer genes tend to have higher degrees in the network. We furthermore observe an increased correlation between degree and MutSig P-values for cancer genes (Pearson correlation: 0.17) compared with unlabeled genes (Pearson correlation: 0.10), as can be seen in Figure 2c and d. Although this indicates that MutSig can identify the highly mutated cancer genes, there exist many well-established cancer genes whose mutation rates lie within the background distribution (i.e. their MutSig P-values are undistinguishable between cancer genes and unlabeled genes). This poses three challenges that have to be addressed: (i) we do not have a high-quality negative class, i.e. in general any gene not classified as a cancer gene might potentially be a cancer driver, (ii) the dataset is imbalanced, a fact that requires attention during supervised classification and (iii) the dataset is affected by knowledge contamination. We address challenges (i) and (ii) with an elaborate and unbiased cross-validation procedure to train and test a classifier, as well as to predict cancer driver genes from the unlabeled genes. The third challenge is addressed by using the moments in the MoPro embeddings. Although we observe that moments such as skewness and kurtosis exhibit positive correlations with the node-degree, this is the case for both, cancer genes and unlabeled genes (see Supplementary Fig. S1).

3.2 Experimental setup

3.2.1 Cross-validation for one-class, imbalanced learning

To address the above mentioned challenges imposed by the class imbalance and the lack of a negative class, we developed a cross-validation procedure that is based on the repeated undersampling of the majority class. The cross-validation procedure is illustrated in Figure 3, and a pseudocode can be found in the Supplementary Algorithm S1. The dataset can be represented as a matrix $D\in\mathbb{R}^{11\,449\times d}$, where d is the number of node features (Fig. 3a). The cross-validation procedure consists of three main steps:

Step 1: Data splits. We split the dataset D into two disjoint datasets, D_l and D_u (Fig. 3b), where D_l consists of all genes in the positive class, and a random subsample of the unlabeled genes. We undersample the majority class such that 10% of samples in D_l are cancer genes. For the sake of training a classifier, we assign the unlabeled samples in D_l to the negative class, and assume this to be the ground truth for the current split. This dataset will be used in the second step to train and evaluate a classifier. The genes in D_u remain unlabeled, and we use the classifier trained on D_l to predict their cancer status.

Step 2: Training and evaluation of the classifier. Next, the dataset D_l is split into a cross-validation (80% of data) and a hold-out test set (20% of data; Fig. 3c). On the cross-validation set, we do a 5-fold stratified cross-validation to find the best hyperparameters of the classifier $\mathcal{C}$, resulting in $\mathcal{C}'$. We retrain the classifier on the complete cross-validation set, and evaluate the predictive performance of $\mathcal{C}'$ on the hold-out test set. Importantly, the cross-validation and hold-out test sets are disjoint. This implies that samples in the hold-out test set have never been seen during training, nor were they used to choose the best hyperparameters of the classifier. This set is solely used to evaluate the ability of the classifier to generalize to unseen samples. The strict separation of the cross-validation and the hold-out test set is necessary to avoid an inflation of the evaluation metrics. Furthermore, each classifier was run and evaluated on the test data only once.

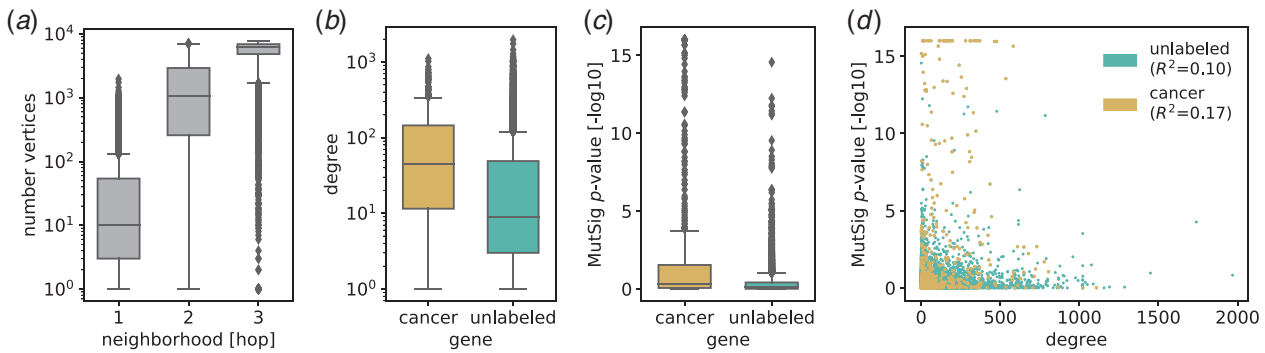

Fig. 2. Dataset description: (**a**) the distribution of neighborhood sizes, for neighborhoods defined as in Equation (1), for $k=1,2,3$. (**b**) The distribution of the node degree, shown for 635 cancer genes and 10 816 unlabeled genes. (**c**) The distribution of the MutSig P-values, in cancer genes and unlabeled genes. (**d**) The correlation between the degree and the MutSig P-values for cancer genes and for unlabeled genes

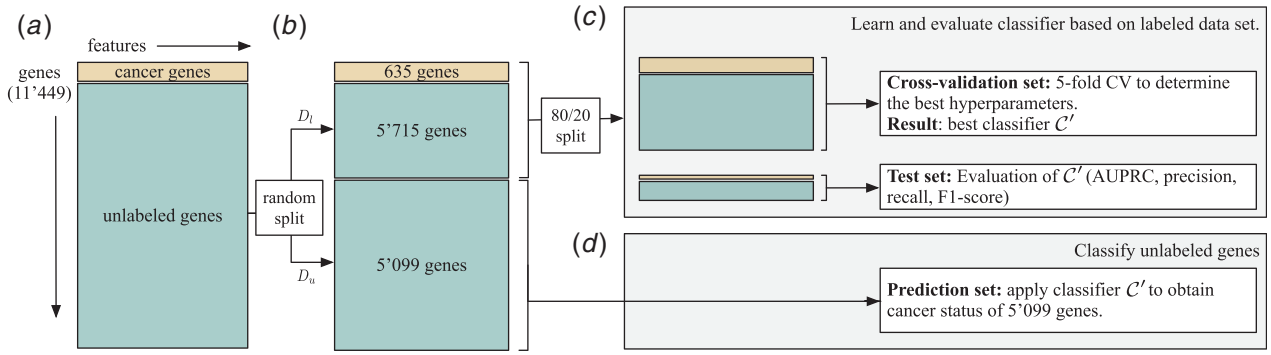

Fig. 3. (**a**) The dataset consists of 11 449 genes, each one represented by a set of features. 635 of those genes are classified as cancer genes (Sondka *et al.*, 2018), highlighted in yellow. The remaining 10 814 genes are unlabeled (green). Cross-validation scheme on *one* data split, resulting in one best classifier $\mathcal{C}'$. (**b**) The unlabeled genes are sub-sampled at random and combined with the cancer genes, giving rise to the labeled dataset D_l. The unlabeled genes in D_l are assigned a negative class label. The remaining unlabeled genes make up the set D_u. Those are the genes for which the cancer status will be predicted in the current split. (**c**) The D_l dataset is split, and 80% of the data are used to find the best hyperparameters of the classifier via 5-fold CV, resulting in the classifier $\mathcal{C}'$. The remaining 20% are used as test set for evaluation of the classifier $\mathcal{C}'$. (**d**) The classifier $\mathcal{C}'$ that has been trained on the cross-validation set in D_l is used to predict the cancer status of genes in D_u

Step 3: Prediction.. Last, we apply the classifier $\mathcal{C}'$ from the previous step to predict the cancer status of genes in the unlabeled dataset D_u (Fig. 3d).

This cross-validation procedure learns to distinguish cancer genes from a random split of the unlabeled genes. However, this is a potentially incorrect assumption, since unlabeled genes in the set D_l might be yet-to-discover cancer genes. For this reason, we repeat the complete cross-validation procedure for r different random splits of the dataset into a labeled subset D_l and a unlabeled subset D_u, resulting in a *set of classifiers* $\mathcal{E} = \{\mathcal{C}'_1, .., \mathcal{C}'_r\}$. Each D_l is divided into an 80% cross-validation dataset and a 20% test dataset. The cross-validation dataset is used for hyperparameter optimization of the classifiers and the test dataset is used for evaluation in terms of area under the precision recall curve (AUPRC), precision, recall and F1-score. $\mathcal{C}'_i$ is the best classifier on the ith random split of the data, determined on the cross-validation dataset. For each $\mathcal{C}'_i \in \mathcal{E}$, where $i = 1, .., r$, we compute the performance metrics (AUPRC, precision, recall, F1-score) on the test set of the ith split, and report the mean and standard deviation of the metrics across all r classifiers in the set $\mathcal{E}$ as the final result.

In order to obtain comparable results for different classification algorithms, we ensure that each algorithm is trained and evaluated on the same r splits of the data. We determine the total number of splits r based on the minimum number of predictions we want to obtain for every gene without a cancer label in the dataset. We set this value to five, resulting in $r = 11$ data splits, and evaluate the effect of varying r in terms of the average AUPRC on the test sets in an experiment (see Section 3.3.3 and Fig. 4b).

Since this splitting of the data is random, the underlying data distribution of the negative class varies from split to split, and the classifier optimized on each split learns the data modalities of the negative class in the current split. Every gene is predicted with each classifier in our set of classifiers (excluding the ones for which it was in the D_l set used for training), and might be classified as a cancer gene by some of the classifiers, but not by others. This can be interpreted as the fact that a gene might be more similar to a cancer gene in some aspects, but more similar to a non-cancer gene in others. Eventually, a gene is classified as a cancer gene according to the majority vote across all classifiers (for which it was not in the training data). In the case of ties, we resort to the conservative prediction of 'no cancer gene'. Importantly, as there exist no known labels for those genes, we analyze them qualitatively.

3.2.2 Classification

We represent each node in the network by the MoPro embeddings computed from the log-transformed MutSig P-values, as described in Section 2.2.3. We apply four different state-of-the-art classification algorithms to predict the binary class labels in the cross-validation procedure described above, using python's sklearn

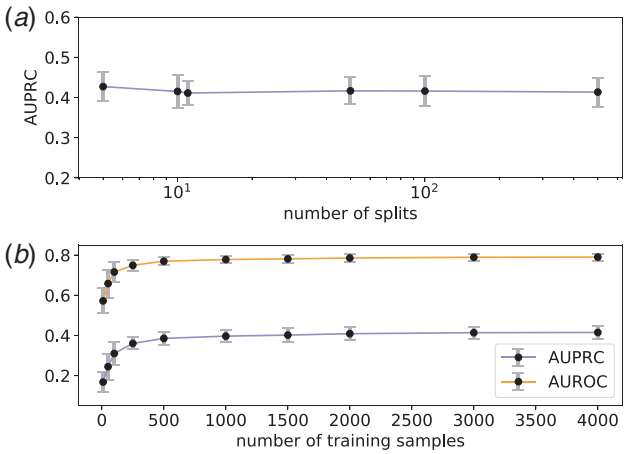

Fig. 4. Evaluation of a set of logistic regression classifiers (hyperparameters as in Table 2). (**a**) AUPRC when varying the number of random splits r, and therefore the number of classifiers in the set $\mathcal{E}$. (**b**) Evaluation metrics as functions of the training set size

module: logistic regression, random forests, support vector machines (SVMs) and gradient boosting. For every classifier, we optimize across a grid of standard hyperparameters, as well as the following data-specific hyperparameters: (i) whether or not to include a scaling step in the classification pipeline (SCALE), (ii) whether to use edge weights to generate node embeddings (WEIGHT), (iii) how to represent weights between two nodes in a k-hop neighborhoods with $k > 1$ (PATH; see Section 2.2.4), as well as (iv) the number of propagation steps t and (v) the number of k-hops to generate moment embeddings from, where $k \in \{1, 2\}$, $t \in [1, \ldots, 6]$. We restrict the value of k to a maximum of 2, as we observe that three-hop neighborhoods in the InBio Map network already span major parts of the network (see Fig. 2a). In order to weight the contribution of node features in local neighborhoods during the generation of moment embeddings (function $f(\cdot, \cdot)$ in Equation 9), we use a simple multiplication between the node features and the edge weights, resulting in a lowering of the contribution of the $-\log_{10}$ transformed P-values for edges that exhibit confidences below 1.0.

In order to evaluate the predictive performance of each classifier, we use the AUPRC, as well as precision, recall and the F1-score. As described in the previous Section, to evaluate the performance of a set of classifiers $\mathcal{E}$ resulting from the cross-validation procedure, those metrics correspond to the average across the classifiers in $\mathcal{E}$. We furthermore report the number of predicted genes that are novel, i.e. those that are not contained in the COSMIC CGC gene set. We would like to note that, although the area under the ROC curve

(AUROC) is a common metric to evaluate binary classifiers, its interpretation is difficult in our setting. Due to the high class imbalance and our primary interest in detecting members of the minority class (i.e. the cancer genes), measuring precision and recall on the minority class is a better suited metric for our task. The AUROC values can be found in the Supplementary Tables S1 and S2.

3.3 Classification of cancer genes

3.3.1 Baseline methods

We compare our approach against univariate baselines, that is we determine the cancer driver status of a gene based on (i) its degree, (ii) its MutSig *P*-value and (iii) its NetSig *P*-value. Those results are listed in Table 1. For all node features, we compare a ranking of the genes by the respective feature (*ranking*) and a prediction with a set of logistic regressors (*LogReg*), generated with the cross-validation procedure described in Section 3.2.1. When ranking the genes based on features, we report precision and recall at the threshold that gave the best F1-score. For the MutSig and NetSig *P*-values, we also evaluate predictive performance after Benjamini–Hochberg (BH) correction at a false discovery rate of 10%. Note that since there is only one classification result for the ranking and Benjamini–Hochberg procedure, there are no standard deviations reported in the table. Furthermore, in both cases, the number of novel genes does not correspond to a majority vote, but is based on a single prediction, using the prediction threshold that resulted in the highest F1-score. We also apply hierarchical HotNet (Reyna *et al.*, 2018), a state-of-the-art network propagation method for the detection of altered subnetworks in cancer, to the MutSig *P*-values (after $-\log_{10}$ transformation). We chose the score permutation scheme to obtain a measure of significance ($P = 0.01$) and report all genes in subnetworks of sizes > 1 as positives. For all methods we observe that using the set of classifiers improves predictive performance with respect to the AUPRC. Furthermore, we observe that the degree of a gene in the network reaches AUROC values of up to 70% (see Supplementary Table S1a), hinting toward the problem of knowledge bias in biological networks. That is, the degree operates as a confounder in those networks.

3.3.2 Cancer gene classification with MoPro embeddings

We generate MoPro embeddings from the $-\log_{10}$ transformed MutSig *P*-values, and use these embeddings as input to the classifiers. The results are listed in Table 2. We evaluate the four classifiers on a grid of hyperparameters, and list the best values of the ones specific to our proposed approach (see Section 3.2.2) in the table. We observe a similar performance of all classifiers with respect to AUPRC, with a minor exception for the random forest classifier. The classification using MoPro embeddings combined with the cross-validation procedure to handle imbalanced classes clearly outperforms the baselines. The baseline with the best AUPRC is the logistic regression classification using MutSig *P*-values (AUPRC = 31.2%). With the MoPro embeddings, AUPRC values of up to 43.7% are achieved with the gradient boosting classifier (closely followed by logistic regression and SVMs). A similar trend can be observed for AUROC scores (see Supplementary Table S1).

For all analyses, we fixed the recall at 23.5%, that is the recall achieved by ranking the NetSig *P*-values. We observe that with MoPro embeddings, we obtain an up to three-fold improvement of precision at that same recall value compared with the NetSig approach (ranked NetSig *P*-values: 21.9%, gradient boosting 63.6%). When contrasting the precision of MoPro embeddings with the one of the best baseline, that is logistic regression using the MutSig *P*-value, we observe an improvement of ∼8%.

We optimize the data-specific hyperparameters, and find that for all classifiers, using at least three propagation steps enables best classification. All methods but random forests performed best when deriving moments from the $k = 2$-hop neighborhoods. Although random forests and gradient boosting achieve better classification performance when using weighted neighborhood distributions, this was not the case in logistic regression and SVMs. There seems to be no clear winner between the generation of weights in *k*-hop

Table 1. Results of cancer gene classification for the baselines

Feature	Method	AUPRC	Precision	Recall	F1	No. of novel
Degree	Ranking	0.096	0.105	0.436	0.169	2368
Degree	LogReg	0.199 (0.007)	0.243 (0.012)	0.236 (0.000)	0.239 (0.006)	905
MutSig	Ranking	0.248	0.474	0.202	0.283	142
MutSig	LogReg	**0.312 (0.007)**	0.552 (0.060)	0.236 (0.000)	0.330 (0.011)	243
MutSig	BH	0.248	0.490	0.191	0.274	126
MutSig	Hier. HotNet	—	0.137	0.111	0.123	444
NetSig	Ranking	0.158	0.219	0.235	0.226	532
NetSig	LogReg	0.275 (0.012)	0.278 (0.019)	0.228 (0.000)	0.250 (0.008)	704
NetSig	BH	0.158	0.263	0.169	0.205	300

Note: The first column indicates the feature that was used to represent each gene during classification, the second column indicates the method that was used for classification. In case of LogReg, we used the cross-validation procedure described in Section 3.2.1 and fixed the recall at 23.5%. AUPRC is the area under the precision recall curve, the method with the highest AUPRC is printed in bold. The last column indicates the number of de novo cancer genes, i.e. those genes that are not contained in the set of cancer genes.

Table 2. Results of cancer gene classification for the moment propagation embeddings

Method	Scale	Weight	Path	*t*	*k*	AUPRC	Precision	Recall	F1	No. of novel
LogReg	True	—	–	3	2	0.434 (0.014)	0.572 (0.046)	0.236 (0.000)	0.334 (0.009)	202
SVM	False	—	–	6	2	0.431 (0.012)	0.584 (0.058)	0.236 (0.000)	0.336 (0.010)	198
RandFor	True	standard	Mean	3	1	0.396 (0.021)	0.560 (0.057)	0.234 (0.004)	0.330 (0.011)	193
GradBoost	True	standard	Max	3	2	**0.437 (0.020)**	0.636 (0.088)	0.236 (0.000)	0.343 (0.012)	150

Note: Classification results for different classifiers using the proposed moment propagation embeddings and the described cross-validation procedure. The Columns 2–6 indicate the hyperparameters that gave the best classification performance for each set of classifiers. *t* and *k* are the hyperparameters of the moment propagation embeddings, namely the number of propagation steps and the neighborhood degree up to which moments are computed. AUPRC is the area under the precision recall curve, the method with the highest AUPRC is printed in bold. The last column indicates the number of de novo cancer genes, i.e. those genes that are not contained in the set of cancer genes.

neighborhoods when using the mean or the maximum aggregation (Section 2.2.4).

3.3.3 Dependence of results on cross-validation parameters

The results in Table 2 are produced with $r = 11$ splits of the data into D_l and D_u (Section 3.2.1). We evaluate the performance of classification with MoPro embeddings for values of r in the range [5, 500] while keeping all data-specific hyperparameters (as described in Section 3.2.2) fixed (using logistic regression). We observe that the classification performance is not affected by changes in the parameter r (see Fig. 4a). Note that we lose ~2% in AUPRC due to fixing the data hyperparameters.

We furthermore evaluate how the size of the training set affects the classification performance. In our proposed cross-validation scheme, the training set size is fixed due to the 5-fold cross-validation, and contains 4064 genes (the cross-validation set contains 5080 genes, such that 4064 genes are used in a 5-fold cross-validation to train the classifier). We conducted an experiment where this number is reduced, ranging between 10 and 4000 samples. We observe a steep increase in performance with an increasing training set size up to 1000 training samples (see Fig. 4b), and a saturation when using more than 1000 samples. This indicates, that at least 1000 samples are required to represent the data distribution during classification.

3.3.4 Ablation study

We conduct an ablation study to understand how the individual parts in the moment propagation embedding contribute to the improved performance. The results can be found in Table 3. We evaluate two different types of experiments: (i) we only use the node feature (i.e. the $-\log_{10}$ transformed MutSig P-value), and propagate it through the network with the propagation embedding $\rho^t(\cdot)$ described in Section 2.2.2. This representation of the node features is used to train a set of logistic regression classifiers. The results of this analysis are listed in the row with label 'propagation only'. (ii) We represent each node with the moment embedding $\eta^k(\cdot), k = 2$ described in Section 2.2.1, without propagating the resulting representation through the network. The results of this approach are listed in the row with label 'moments only'. We observe that removing the moment embedding results in a severe drop in performance of ~8%, while keeping moments but not propagating them leads to a less severe reduction (~2.8%). This observation indicates that the main improvement of performance compared with the baselines is due to the description of a node by means of the distribution of node features in its neighborhood, motivating the development of methods that improve the representation of local neighborhoods.

3.4 Evaluation of predicted cancer driver genes

To generate a candidate gene list, we created a consensus set of all genes as follows: for each of the four classification algorithms (logistic regression, SVM, random forest, gradient boosting), we took the intersection of genes that were classified as 'novel' (see Table 2). This means that a gene in the consensus set has (i) not been classified as a cancer driver gene in the CGC dataset, and (ii) all four sets of classifiers identified the gene as a cancer driver (as described in Section 3.2.1). This led to 50 candidate genes, 31 of which were significant in the MutSig data ($P = 8.04 * 10^{-42}$, hypergeometric test), 10 in the NetSig data ($P = 1.07 * 10^{-6}$, hypergeometric test) and 12 with hierarchical HotNet ($P = 2.04 * 10^{-7}$, hypergeometric test), where P-values measure whether the set of 50 consensus genes is significantly enriched with MutSig, NetSig and hierarchical HotNet hits, respectively. By removing all genes from the consensus set that were detected with at least one other method, 14 novel genes remained. For those, we performed a literature review to estimate the evidence for links to cancer.

Table 3. Results of the ablation study for the set of logistic regression classifiers

Setting	Method	AUPRC
Baseline	LogReg	0.434 (0.014)
Propagation only	LogReg	0.348 (0.010)
Moments only	LogReg	0.406 (0.011)

Note: In *propagation only*, the node feature is propagated, but no moment embedding is computed. In *moments only*, moments are computed, but no propagation embedding is computed. The first row repeats the baseline results (moment propagation embeddings) for comparability reasons.

In brief, four genes have a direct link to tumorigenesis in human. The transcription factor GATA4 is a known tumor suppressor in Glioblastoma Multiforme (Agnihotri *et al.*, 2011). . In breast cancer patients, ID2 is upregulated in brain metastasis and high expression is linked to an increased risk of developing relapse (Kijewska *et al.*, 2019). Last, FOS exhibits recurrent rearrangements Osteoblastoma (Fittall *et al.*, 2018).

Five genes can be strongly linked to tumor relevant behavior and pathways. ACVR1B (also known as ALK4) is linked to tumorigenesis through its interaction with activin-A (Kalli *et al.*, 2019; Rautela *et al.*, 2019). CASP10 inhibition leads to reduced apoptosis, while loss-of-function of RAP1A causes a reversion to a non-malignant phenotype in a model of invasive carcinoma (Stammer *et al.*, 2017). MYLK is involved in proliferation and the migration of cancers of the breast, prostate and colon (Avizienyte *et al.*, 2005; Cui *et al.*, 2010; Zhou *et al.*, 2008). CSNK1A1, a member of the CK1 kinase family, is a regulator of the autophagic pathway in RAS-driven cancers, and knock-out experiments lead to cell death in Multiple myeloma (Carrino *et al.*, 2019; Cheong *et al.*, 2015).

For the remaining six genes, five (CASP1, CASP14, RBL1, HNF4A and RALA) had weaker links (e.g. expression linked or pathway membership), but no clear experimental evidence (Gouravani *et al.*, 2020; Krajewska, 2005; Schade *et al.*, 2019; Seibold *et al.*, 2019; Wang *et al.*, 2020).

Only for one gene (DLGAP2) we could not find any evidence for a link to cancer.

4 Discussion and conclusions

In this article, we proposed a novel approach for the identification of cancer driver genes by integrating MutSig summary statistics (Lawrence *et al.*, 2014) with PPI networks (Lage *et al.*, 2007; Li *et al.*, 2017). In stark contrast to state-of-the-art approaches that set out to solve this problem with unsupervised processes, we developed an innovative node embedding procedure (MoPro embeddings) to enable supervised classification of cancer driver genes. Reformulating the problem of cancer-gene prediction in a supervised fashion enables learning from *what we already know*: we include knowledge on the data distributions of well-established cancer driver genes and learn from these distributions to improve the prediction task. We do so by combining two concepts: (i) the representation of a node based on the distribution of node features in its k-hop neighborhood, followed by (ii) a network propagation. The neighborhood distributions in (i) are described concisely by their first four moments, which constitutes a computationally efficient summary, and addresses the knowledge contamination that often confounds analyses of biological networks.

We show that our approach outperforms baselines with respect to AUPRC by a margin of more than 10%, and that results are stable with respect to the hyperparameters of the method. Interestingly, we find that the main improvement in predictive performance is presumably caused by the representation of the distributions of node features in a gene's neighborhood, rather than the network propagation. This finding paves the way for further research: while the proposed representation of the distributions by means of their moments is straightforward and computationally efficient, another option is to exploit principles of optimal transport (Villani, 2008) to compare two nodes based on the distributions of features in their neighborhoods. Togninalli *et al.* (2019) developed a kernel based on Wasserstein distances between distributions for graph classification, and this idea can be readily extended to node classification. Another

possible route for future research is to build models that combine both, node embeddings and classification, and to train them end-to-end, as is done with graph convolutional networks (e.g. Duvenaud *et al.*, 2015; Hamilton *et al.*, 2017; Kipf and Welling, 2016).

The set of high confidence consensus genes discovered with our proposed approach contained both, genes that were previously identified as cancer drivers with methods such as MutSig, NetSig and hierarchical HotNet, as well as novel genes that were not detected with established methods. Those genes constitute promising targets for future biological evaluation, and their detection showcases the potential of combining network-derived features with supervised machine learning techniques for the prediction of cancer driver genes.

Acknowledgements

The results shown here are in whole or part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga.

Funding

This work was supported by the SNSF Starting Grant 'Significant Pattern Mining' (no. 155913, K.B.).

Conflict of Interest: none declared.

References

Agnihotri,S. *et al.* (2011) A GATA4-regulated tumor suppressor network represses formation of malignant human astrocytomas. *J. Exp. Med.*, **208**, 689–702.

Avizienyte,E. *et al.* (2005) The SRC-induced mesenchymal state in late-stage colon cancer cells. *Cells Tissues Organs.*, **179**, 73–80.

Bailey,M.H. *et al.* (2018) Comprehensive characterization of cancer driver genes and mutations. *Cell*, **173**, 371–385.e18.

Carrino,M. *et al.* (2019) Prosurvival autophagy is regulated by protein kinase CK1 alpha in multiple myeloma. *Cell Death Discov.*, **5**, 98.

Cheong,J.K. *et al.* (2015) Casein kinase 1α-dependent feedback loop controls autophagy in RAS-driven cancers. *J. Clin. Invest.*, **125**, 1401–1418.

Cowen,L. *et al.* (2017) Network propagation: a universal amplifier of genetic associations. *Nat. Rev. Genet.*, **18**, 551–562.

Cui,W.-J. *et al.* (2010) Myosin light chain kinase is responsible for high proliferative ability of breast cancer cells via anti-apoptosis involving p38 pathway. *Acta Pharmacol. Sin.*, **31**, 725–732.

Duvenaud,D.K. *et al.* (2015) Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in Neural Information Processing Systems*, Curran Associates Inc, USA, Vol. 28. pp. 2224–2232.

Fittall,M.W. *et al.* (2018) Recurrent rearrangements of FOS and FOSB define osteoblastoma. *Nat. Commun.*, **9**, 2150.

Gilmer,J. *et al.* (2017) Neural message passing for quantum chemistry. In: *Proceedings of the 34th InternatUSAional Conference on Machine Learning-Volume* **70**, PMLR, USA, pp. 1263–1272. JMLR. org.

Gouravani,M. *et al.* (2020) The NLRP3 inflammasome: a therapeutic target for inflammation-associated cancers. *Expert Rev. Clin. Immunol.*, **16**, 175–187.

Hamilton,W. *et al.* (2017) Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems*, Curran Associates Inc, USA, Vol. 30. pp. 1024–1034.

Hanahan,D. and Weinberg,R.A. (2011) Hallmarks of cancer: the next generation. *Cell*, **144**, 646–674.

Horn,H. *et al.* (2018) NetSig: network-based discovery from cancer genomes. *Nat. Methods*, **15**, 61–66.

Hristov,B.H. *et al.* (2020) A guided network propagation approach to identify disease genes that combines prior and new information. *arXiv preprint arXiv : 2001.06135*.

Jia,P. *et al.* (2011) dmGWAS: dense module searching for genome-wide association studies in protein–protein interaction networks. *Bioinformatics*, **27**, 95–102.

Kalli,M. *et al.* (2019) Activin a signaling regulates IL13Rα2 expression to promote breast cancer metastasis. *Front. Oncol.*, **9**, 32.

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

Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.

Kijewska,M. *et al.* (2019) Using an in-vivo syngeneic spontaneous metastasis model identifies ID2 as a promoter of breast cancer colonisation in the brain. *Breast Cancer Res.*, **21**, 4.

Kipf,T.N. and Welling,M. (2016) Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krajewska,M. (2005) Tumor-associated alterations in caspase-14 expression in epithelial malignancies. *Clin. Cancer Res.*, **11**, 5462–5471.

Lage,K. *et al.* (2007) A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nat. Biotechnol.*, **25**, 309–316.

Lawrence,M.S. *et al.* (2014) Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, **505**, 495–501.

Leiserson,M.D. *et al.* (2015) Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.*, **47**, 106–114.

Li,T. *et al.* (2017) A scored human protein–protein interaction network to catalyze genomic interpretation. *Nat. Methods*, **14**, 61–64.

Li,T. *et al.* (2018) GeNets: a unified web platform for network-based genomic analyses. *Nat. Methods*, **15**, 543–546.

Mularoni,L. *et al.* (2016) OncodriveFML: a general framework to identify coding and non-coding regions with cancer driver mutations. *Genome Biol.*, **17**, 128.

Niu,Y. *et al.* (2017) PrePhyloPro: phylogenetic profile-based prediction of whole proteome linkages. *PeerJ*, **5**, e3712.

Rautela,J. *et al.* (2019) Therapeutic blockade of activin-A improves NK cell function and antitumor immunity. *Sci. Signal.*, **12**, eaat7527.

Reyna,M.A. *et al.* (2018) Hierarchical hotnet: identifying hierarchies of altered subnetworks. *Bioinformatics*, **34**, i972–i980.

Rossin,E.J. *et al.*; International Inflammatory Bowel Disease Genetics Constortium. (2011) Proteins encoded in genomic regions associated with immune-mediated disease physically interact and suggest underlying biology. *PLoS Genet.*, 7, e1001273.

Ruffalo,M. *et al.* (2015) Network-based integration of disparate omic data to identify 'silent players' in cancer. *PLoS Comput. Biol.*, **11**, e1004595.

Sanchez-Garcia,F. *et al.* (2014) Integration of genomic data enables selective discovery of breast cancer drivers. *Cell*, **159**, 1461–1475.

Schade,A.E. *et al.* (2019) RB, p130 and p107 differentially repress G1/S and G2/M genes after p53 activation. *Nucleic Acids Res.*, **47**, 11197–11208.

Seibold,M. *et al.* (2019) RAL GTPases mediate multiple myeloma cell survival and are activated independently of oncogenic RAS. *Haematologica*.doi: 10.3324/haematol.2019.223024.

Shervashidze,N. *et al.* (2011) Weisfeiler-Lehman graph kernels. *J. Mach. Learn. Res.*, **12**, 2539–2561.

Sondka,Z. *et al.* (2018) The cosmic cancer gene census: describing genetic dysfunction across all human cancers. *Nat. Rev. Cancer*, **18**, 696–705.

Song,Z. *et al.* (2019) HIF-1α-induced RIT1 promotes liver cancer growth and metastasis and its deficiency increases sensitivity to sorafenib. *Cancer Lett.*, **460**, 96–107.

Stammer,R.M. *et al.* (2017) Synergistic antitumour properties of viscumTT in alveolar rhabdomyosarcoma. *J. Immunol. Res.*, **2017**, 1–13.

Szklarczyk,D. *et al.* (2019) STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.*, **47**, D607–D613.

Togninalli,M. *et al.* (2019) Wasserstein Weisfeiler-Lehman graph kernels. In: *Advances in Neural Information Processing Systems*, vol. **32**, pp. 6436–6446.

Vandin,F. *et al.* (2011) Algorithms for detecting significantly mutated pathways in cancer. *J. Comput. Biol.*, **18**, 507–522.

Vandin,F. *et al.* (2012) Discovery of mutated subnetworks associated with clinical data in cancer. In: *Biocomputing 2012*, pp. 55–66. World Scientific Publishing Co. Pte. Ltd., Singapore.

Villani,C. (2008) *Optimal Transport: Old and New*, Vol. **338**. Springer Verlag, Berlin, Heidelberg.

Vogelstein,B. *et al.* (2013) Cancer genome landscapes. *Science*, **339**, 1546–1558.

Wang,Z. *et al.* (2020) Nuclear receptor HNF4α performs a tumor suppressor function in prostate cancer via its induction of p21-driven cellular senescence. *Oncogene*, **39**, 1572–1589.

Weisfeiler,B. and Andrei,A.L. (1968) A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, **2**, 12–16.

Willsey,A.J. *et al.* (2013) Coexpression networks implicate human midfetal deep cortical projection neurons in the pathogenesis of autism. *Cell*, **155**, 997–1007.

Zhou,X. *et al.* (2008) Myosin light-chain kinase contributes to the proliferation and migration of breast cancer cells through cross-talk with activated ERK1/2. *Cancer Lett.*, **270**, 312–327.