Computers in Biology and Medicine 114 (2019) 103362

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Cancer driver gene discovery in transcriptional regulatory networks using influence maximization approach

Majid Rahimi[a], Babak Teimourpour[a,*], Sayed-Amir Marashi[b]

[a] Department of Information Technology Engineering, Faculty of Industrial and Systems, Tarbiat Modares University, Tehran, Iran
[b] Department of Biotechnology, College of Science, University of Tehran, Tehran, Iran

ARTICLE INFO

Keywords:
Cancer driver gene
Cancer genomics
Transcriptional regulatory network
Gene expression data
Influence maximization

ABSTRACT

Cancer driver genes (CDGs) are the genes whose mutations cause tumor growth. Several computational methods have been previously developed for finding CDGs. Most of these methods are sequence-based, that is, they rely on finding key mutations in genomic data to predict CDGs. In the present work, we propose iMaxDriver as a network-based tool for predicting driver genes by application of influence maximization algorithm on human transcriptional regulatory network (TRN). In the first step of this approach, the TRN is pruned and weighted by exploiting tumor-specific gene expression (GE) data. Then, influence maximization approach is used to find the influence of each gene. The top genes with the highest influence rate are selected as the potential driver genes. We compared the performance of our CDG prediction method with fifteen other computational tools, based on a benchmark of three different cancer types. Our results show that iMaxDriver outperforms most of the state-of-the-art algorithms for CDG prediction. Furthermore, iMaxDriver is able to correctly predict many CDGs that are overlooked by all previously published tools. Due to this relative orthogonality, iMaxDriver can be considered as a complementary approach to the sequence-based CDG prediction methods.

1. Introduction

1.1. Importance of CDG discovery

The quest for key mutated genes which are related to the cancer has been conducted by several recent researches [1–3]. The main idea behind current cancer driver gene (CDG) discovery methods is the assumption that frequent mutations in certain genes causes the cancer. Not all mutations in a cancer genome are related to cancer. Consequently, various computational methods are used to distinguish cancer driver mutations from passenger mutations.

The available methods for CDG discovery mainly rely on genomic and/or transcriptomic data. For example, DawnRank [4] tries to find driver genes using mutational and transcriptome data together with molecular interaction network information. ActiveDriver [3] exploits the data about post-translational modification sites of proteins that mutated in cancer genomes. Likewise, e-Driver [5] tries to find biased mutation rate in functional regions of a protein. Similarly, the method proposed by Youn and Simon [6] accounts for the functional impact of mutations on proteins for finding CDGs. Moreover, OncodriveFM [7] and OncodriveCLUST [8] present methods which rely on assessing the functional impact of cancer genome variants on proteins. Dendrix [9], MeMo [10], MSEA [11] and CoMDP [12] use mutation profiles to identify cancer driver "pathways". On the other hand, DriverNet [13] tried to find driver mutations by evaluating their effect on transcriptome. In contrast, for finding CDGs, NetBox [14] considers both protein-protein interactions and signaling pathways in order to find "network modules" which are affected by mutations. MutsigCV [15] exploits exome sequences to detect the heterogeneities in cancer datasets, and then, CDGs are determined based on their mutation frequency in different cancer types. Finally, MDPFinder [16] and iPAC [17] combine gene expression and mutational data to identify CDGs.

There are some limitation and shortcomings in the currently available methods. These methods have high rate of false positive CDGs resulting low precision and F-measure. Furthermore, all of these methods heavily rely on mutation data, which is noisy, and additionally, may not be always available with desired quality. Moreover, most of CDGs found by each of these methods are in common with the set of CDGs found by other methods. Due to these limitations, we propose a new network-based driver gene prediction method. Our approach relies on the structure of gene regulatory network with no requirement of mutation data. This approach exploits an independent

* Corresponding author. Room No. 210, 2nd Floor, School of Systems and Industrial Engineering, Tarbiat Modares University (TMU), Chamran/Al-e-Ahmad Highways. Intersection, Tehran, P.O. Box 14115-111, Iran.
E-mail address: b.teimourpour@modares.ac.ir (B. Teimourpour).

https://doi.org/10.1016/j.compbiomed.2019.103362
Received 25 January 2019; Received in revised form 8 July 2019; Accepted 16 July 2019
Available online 17 July 2019
0010-4825/ © 2019 Elsevier Ltd. All rights reserved.

source of information, and we show that it is able to find many new CDGs, and therefore, can be considered as an orthogonal CDG prediction method. Finally, we show that the performance of our method is better than many of the other existing approaches.

Transcriptional regulation network (TRN) is the fundamental network for controlling cellular processes. Gene regulation controls the activity of the genes at the transcription level. Transcription factors (TF) are key components in the cell that orchestrate the regulations. In other words, a TRN shows how any of the TFs regulates the expression of other TFs and genes. Most diseases, including cancer, are related to some dysfunction of TFs, which indicates the importance of analyzing TFs in biomedical research [18].

In this study, we propose iMaxDriver as a novel network-based method for finding CDGs by applying influence maximization approach on a cancer-specific tailored TRN. We show that iMaxDriver is able to improve the accuracy of CDG prediction, while it correctly finds many CDGs which are not detectable by the current sequence-based approaches.

1.2. Theoretical background

Influence maximization (IM) is a concept widely used in social network analysis to explain diffusion of information in a network [19]. Indeed, the IM problem is finding a minimal subset of nodes (*i.e.*, the seed set) which have the greatest influence on other nodes. This problem is known to be an NP-hard optimization problem [20].

The IM problem is typically modeled by one of the two main approaches, namely, "independent cascade" or "linear threshold". In these models each of the nodes in the network is labeled as "active" or "inactive". At the beginning of the algorithm, a set of nodes are assumed to be active, while others are considered as inactive. Each of the active nodes can activate its (inactive) consecutive node based on a certain criterion, which depends on the IM approach. Independent cascade model only exploits probability value of edges to make decision for node activation, while in linear threshold model, in addition, a threshold value for each node is required.

The linear threshold model considers a threshold θ_v for each node v and a weight $w_{b,v}$ for the incoming edge from node b. The values of θ_v and $w_{b,v}$ should be normalized so that they are in [0,1] interval. Moreover, the sum of weights of incoming edges of any node should not exceed 1:

$$\sum_{\text{edge } x \to v} w_{x,v} \leq 1 \tag{1}$$

In each iteration of the algorithm, the active nodes will activate an inactive node if the sum of weights of incoming edges is at least equal to θ_v:

$$\sum_{\text{active } b} w_{b,v} \geq \theta_v \tag{2}$$

In Fig. 1, the linear threshold method is applied to a small toy model, which can be considered as a fictitious TRN. In this model, the regulatory interactions are weighted and an activation threshold value is specified for each node.

We have used a combination of TRN with gene expression (GE) data as input of linear threshold approach of IM for finding CDGs. Each of the TFs exists in TRN, has been considered as initial active node to find out coverage (number of genes that it activates) of all nodes. Since only TFs in the network have outcome edges, for simplicity we have considered all of nodes as initial active node. In this model an active node can affect inactive nodes, indeed, the activation notion implies affecting downstream genes and showing the flow of GE change due to transcriptional regulation in the network which occurs in the cascading manner.

2. Methods

In this section we firstly describe the iMaxDriver pipeline, which comprises two different steps, namely, network construction and IM algorithm for finding CDGs. Finally, we describe the assessment of this pipeline, together with fifteen other cancer driver-predicting computational tools, based on three of available dataset benchmarks. The overall procedure of iMaxDriver method is shown in Fig. 2.

2.1. Network construction

2.1.1. Transcription regulatory network data

We considered RegNetwork database [21] (http://www.regnetworkweb.org/) as an example of a weighted TRNs to be used as data source of iMaxDriver for weighted networks (iMaxDriver^{W}). In RegNetwork, the list of gene regulatory interactions from different methods and multiple databases is collected. More specifically, we retrieved the human TRN data of RegNetwork obtained from all databases and all methods. It should be noted that RegNetwork, in addition to the TRNs, also contains microRNA regulatory interactions, which are omitted in the present study. The retrieved dataset contains 150,202 TF-gene and TF-TF regulatory interactions. In addition, we exploited the confidence values for assigning weights to interactions which is available in website of RegNetwork as 'CONFIDENCE' column in search page. The edge weighting procedure was done based on the provided confidence of relationships, namely, 'low', 'medium' and 'high'. The 'high' confidence is used for experimentally validated regulations, 'medium' for predicted regulations in more than one method and the rest as 'low' confidence. We assumed 0.2, 0.5 and 0.8 respectively for low, medium and high confidence values as edge weights. According to structure of constructed network in influence maximization model, changing these values only could affect results slightly.

As an example of non-weighted TRNs, we considered the TRRUST database [22] (http://www.grnpedia.org/trrust), to be used as data source of iMaxDriver for non-weighted networks (iMaxDriver^{N}). The TRRUST has been constructed using text mining methods, and then, by manual curation. We retrieved a human TRN from TRRUST v2 which includes with 9,396 TF-gene and TF-TF regulatory interactions.

2.1.2. Gene expression data

We obtained GE data for three specific cancer types from GEO database (Table 1). The GE values were taken from the curated GEO dataset (GDS). In cases where only the.CEL files were available, we extracted the expression values using RMA method implemented in Affy package in R [23]. In each of the selected GEO datasets, GE value is reported for both the tumor and for its adjacent normal tissue. In the preprocessing step, first we removed rows with missing gene ID and then combined gene expression values which their gene ID was synonym by averaging gene expression value of respective columns, afterwards we normalized the GE value e such that $e \in [0,1]$ by dividing each value by maximum the value of GE in the relevant tissue type (tumor or normal).

2.1.3. Weighting the network

In the next step, the collected data were preprocessed to be prepared for IM algorithm. The linear threshold approach in IM problem requires a threshold θ_v value bound to each gene (node) v which stored in list T. For each gene v, we assigned a threshold value as following:

$$\theta_v = |e_{v,n} - e_{v,c}| \tag{3}$$

where $e_{v,n}$ is the GE value of v in the normal tissue and $e_{v,c}$ is its GE value in the corresponding adjacent tumor tissue. Since all of the expression values have been normalized before, $\theta_v \in [0,1]$. In the next step, the edge weights should also be normalized such that the constraint in equation (1) is satisfied. In the iMaxDriver^{W} the edge confidence values are provided, and therefore, we exploited them to

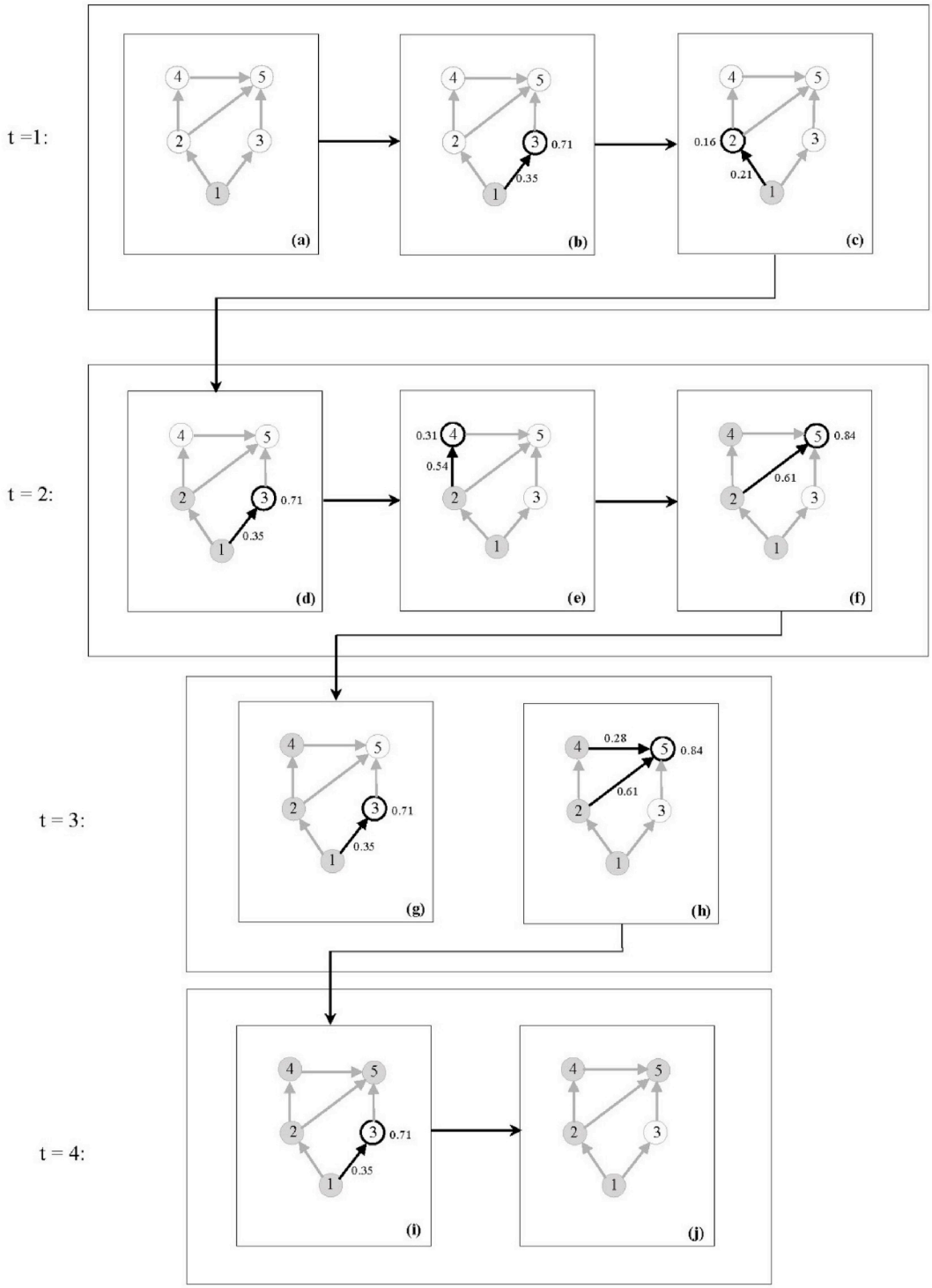

Fig. 1. At the beginning of the algorithm, node 1 is considered as an active node (gray) and other nodes are assumed to be inactive (white) (a). In the next step, the potential influence of the only active node on other nodes should be assessed. In (b) and (c), node 1 tries to activate node 3 and node 2 respectively which successfully activates node 2 but cannot activate node 3 because of insufficient weight of the edge connecting node 1 to node 3. There is no inactive node remained in this iteration (t = 1) with incoming edge from an active node. Note that newly activated nodes cannot activate others in this iteration. Now, since no other node can be activated furthermore, this iteration will be ended. In the next iteration (t = 2) (d), the node 3 cannot be activated, because it only has one incoming edge with insufficient edge weight as assessed in the (b). In (e) node 2 activates node 4 whereas cannot activate node 5 due to insufficient edge weight rather than θ value of node 5 (f). the node 3 cannot be activated as mentioned above (g). In the next iteration (t = 3), the node 4 is active and when the incoming edges of node 5 assessed (h), the sum of weights of incoming edges from node 2 and node 4, exceeds θ value of node 5 and it will be activated in this iteration. In the next iteration (t = 4) (i), the node 3 cannot be activated. Finally, in (j), no node can be activated furthermore and the algorithm will be terminated.

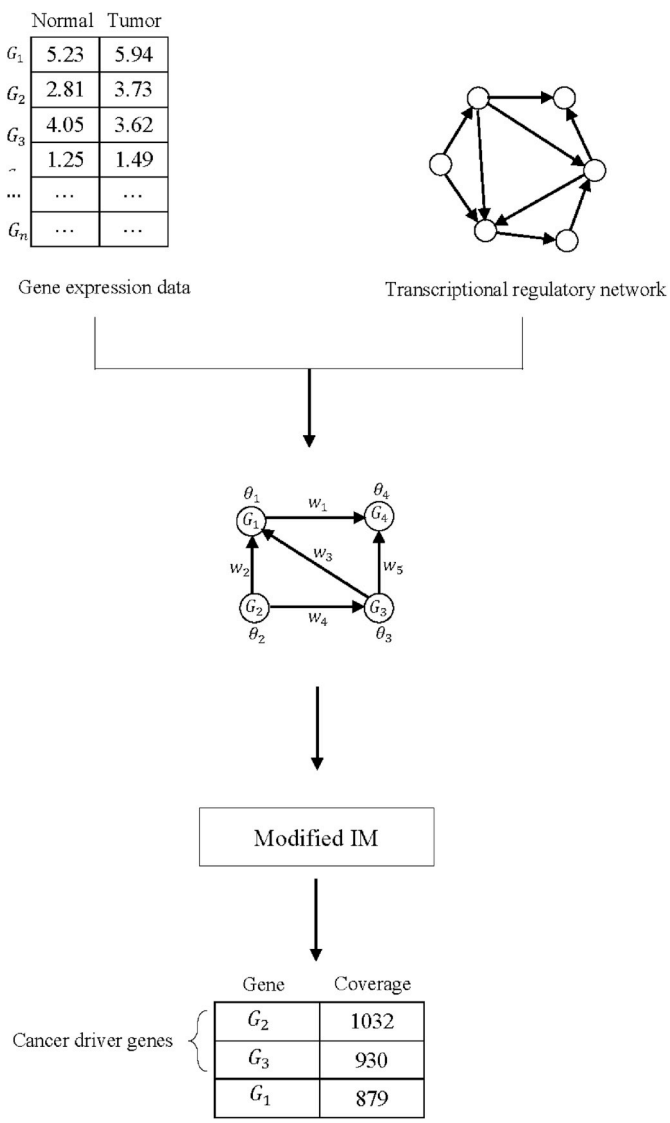

Fig. 2. The overall procedure of iMaxDriver method. At first, the network is constructed using raw TRN with associating threshold values for each node based on their gene expression values and weighting edges randomly or based on TRN edge weights. Finally, by using the modified IM algorithm with the constructed network as input, genes are ranked based on their coverage.

Table 1
The list of GE datasets used in the present study.

GEO ID	Cancer type	Number of entries
GSE3268	Lung squamous cell carcinoma (LUSC)	22,284
GSE32323	Colon adenocarcinoma (COAD)	54,675
GSE15852	Breast invasive carcinoma (BRCA)	19,813

compute the required edge weights by normalizing these values. More specifically, the normalization is done by division of edge weight by the maximum weight of all edge that their destination is same as destination of it:

$$w_{b,v} = w_{b,v}/\max\left(\sum_{\text{edge } x \to v} w_{x,v}\right) \tag{4}$$

When the edge weights are unavailable i.e. original gene regulatory network is presented as a non-weighted graph, one may assign random numbers to put weights on the edges, and then, normalize these weights such that equation (1) is satisfied. equation (4) further is applicable for iMaxDriver^{N}, but since the random weight values are independent, the normalization for iMaxDriver^{N} can be done simpler by division of random edge weight $r_{b,v}$ by the sum of weights of the incoming edges to node v:

$$w_{b,v} = r_{b,v}/\sum_{\text{edge } x \to v} r_{x,v} \tag{5}$$

Indeed, practically both equations generate normalized random values for input of the algorithm, therefore the results will not be changed by using equation (4) or (5). The complete process of the network weighting is shown in Fig. 3.

2.2. The iMaxDriver algorithm

As mentioned above, the IM approach tries to find a minimal subset of initially active nodes with maximum influence. In the classic algorithm, one should test all possible subsets of nodes (*i.e.*, 1-member subsets, 2-member subsets, etc.) as the seed subset. In the present work, we used a modified version of IM which considers only a single active node as the seed. We developed a multi-threaded implementation of the algorithm which could be run on multi-core processors. The implementation of iMaxDriver^{W} and iMaxDriver^{N} algorithms, is available from https://github.com/majid-rh/iMaxDriver.

The network weighting and preprocessing steps are different based on the type of input network. In the case of iMaxDriver^{W} that edge weights are non-random and are based on the provided interaction confidence values, the weights are defined, and therefore, the algorithm is run once (Algorithm 1). Conversely, in the case of iMaxDriver^{N} that network is non-weighted, since the random values can affect final results, we repeated the experiment 100 times and computed the average of all coverage values, to alleviate the effect of random weight assignments (Algorithm 2). The *FindCoverage* function is the main procedure used by Algorithm 1 and Algorithm 2 for computing coverage (Algorithm 3), representing the modified version of the linear threshold approach.

The input of iMaxDriver algorithm is the graph G representing TRN and a list T containing the threshold values. The output of the algorithm is a list of genes sorted by their coverage count in descending manner. In the Algorithm 1, which is provided for weighted graphs, in the first line, we create an empty list of tuples. Each tuple contains the gene name and its coverage count. In line 3, we calculate *MaxIncome* as the maximum of sum of the weights of incoming edges (equation (4)). In lines 4–6, the edge weights are normalized (equation (4)). Afterwards, in lines 7–10, *FindCoverage* function is called for each node v in the G and the returned value of coverage count is stored in *Result*. Finally, in line 11 the *Result* list is sorted by *CoverageCount* in descending order.

In Algorithm 2, which is provided for non-weighted graphs, in line 1, we first create an empty list of tuples, each tuple contains gene name and its coverage (number of genes that activated). In line 3, a loop defined to run the coverage discovery for 100 times and for each of the iterations, in lines 4–6 new random values assigned to the edges. Note that in each iteration the edge weights regenerated randomly to overcome random value effects on final results. Next, for normalization of randomly generated edge weights, in line 7, a *for* loop defined to iterate over all of the nodes and normalize the incoming edges to them. In lines 8–9, the sum of weights of incoming edges of node v is calculated. Now in lines 10–12, for each incoming edge to node v, the edge weights divided by *Sum* value according to equation (5). Next, in lines 14–16, the core function of coverage finding (*FindCoverage*) called for each

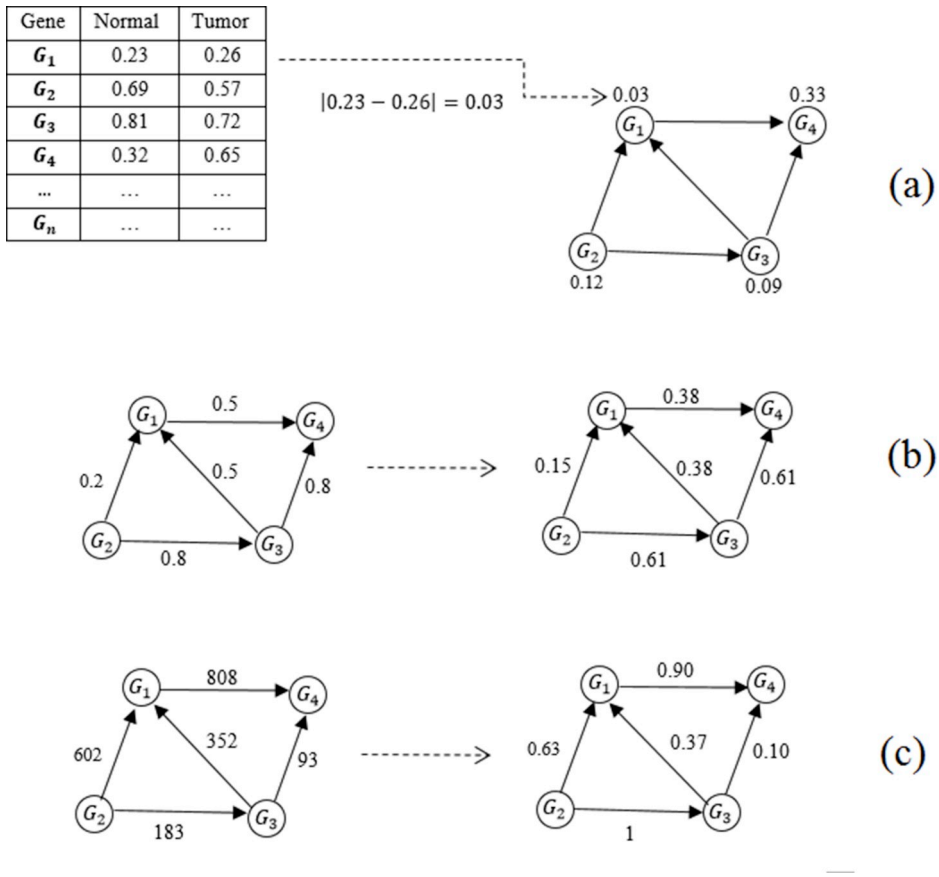

Fig. 3. An example of the network weighting procedure based on a given TRN. (a) For each gene v, the threshold value is computed; (b) In case of weighted TRNs, the weights are normalized, to satisfy equation (1). (c) in case of non-weighted TRNs, the edge weights are random generated, and then normalized, to satisfy equation (1).

node v and the results are integrated by averaging coverage values.

The algorithm of *FindCoverage* function is provided in Algorithm 3. In this algorithm, the weighted graph G and list T are the inputs and coverage count (*ActiveCount*) is output. At the beginning, in line 1, initially the gene v considered as the only active gene and is assigned to *ActiveNodes*. Subsequently, in line 2, the number of currently activated genes by gene v assigned to be zero. The algorithm loops through lines 3–14, until an iteration with no new gene activation. This defined in the algorithm using *while* statement in line 3. In line 4, an empty list of the genes that are going to be activated in this iteration (*NewActiveNodes*), is created. Subsequently, in lines 7–10, for each inactive gene, if difference of sum of incoming edge weights from currently active genes and threshold of the gene, be larger than α, the gene will be activated. Finally, in line 13, the newly activated nodes (*NewActiveNodes*) is added to current active nodes (*ActiveNodes*).

In line 9, we have used α as a parameter in the algorithm for alleviating the differences exist in threshold values and edge weights causing the algorithm to not work properly. We have set $\alpha = 0.15$ for obtaining the best result. The larger value of α causing less genes to be activated at final step of the algorithm and vice versa. If α be too small, all of the genes will be activated and if α be too large, none of the genes will be activated and results in trivial outcomes.

2.3. Evaluation method

We evaluated iMaxDriver by comparing its results with fifteen popular computational CDG prediction methods. The list of genes introduced as CDGs by the fifteen computational methods are obtained from DriverDB v2 [24] for the evaluation with same input for all methods. The details of computational methods used for evaluation is available in Table 2. These lists are reported in DriverDB webpage (http://driverdb.tms.cmu.edu.tw/driverdbv2/) based on three different cancer datasets, namely breast invasive carcinoma (BRCA) [25], lung squamous cell carcinoma (LUSC) [26] and colon adenocarcinoma (COAD) [27], which are parts of the Cancer Genome Atlas (TCGA) [28]. We also retrieved the lists of genes identified as CDG by each of the fifteen methods for benchmarking.

In the next step, we used iMaxDriver for predicting CDGs, and then, for iMaxDriver and the other fifteen methods, we assessed the accuracy of the predicted CDGs by comparing each list with the Cancer Gene Census (CGC) [29] gene list, as the gold standard (available from https://cancer.sanger.ac.uk/census). Next, the IM approach was independently applied on each of the three GE datasets (see Subsection 3.1) to compute the influence of each TF in the network. The results of iMaxDriver are provided for each cancer type as a list of potential CDGs sorted by their influence (coverage count) in descending order. Subsequently, by discretizing the results based on a threshold value we classified the genes either as CDG or non-CDG. For fine-tuning the threshold value used for binary classification, we exploited pROC [30] package in R.

The *F*-measure is a prevalent measure for evaluating the classifiers and is a good measure considering both of precision and recall

Table 2
The details of the computational methods used for comparison with iMaxDriver.

Method Name	Feature(s)	Data type
ActiveDriver [3]	Protein phosphorylation signaling sites	Single nucleotide variants related to phosphorylation
CoMDP [12]	Co-occurred mutated driver pathways	Mutation data
Dendrix [9]	Coverage and exclusivity of mutations	Mutation data
DawnRank [4]	Downstream expression impact in network	GE and network and mutation data
e-Driver [5]	Protein functional region mutation rates	Mutation profiles
DriverNet [13]	Effect of mutations on mRNA network	Mutation data and transcriptional network
iPAC [17]	Copy number alterations	GE and mutation data
MDPFinder [16]	Mutual exclusivity of gene modules	GE and mutation data
MeMo [10]	Genes correlation	Mutation data and network
MSEA [11]	Combination of data associated to disease	Pathways and networks
MutsigCV [15]	Frequency of mutations and spectrum	GE and exome sequence
NetBox [14]	Functional modules in cellular networks	Mutation data and network
OncodriveCLUST [8]	Somatic mutation clustering	Mutation data
OncodriveFM [7]	Functional mutations impact on gene	Somatic variants data
Simon [6]	Mutations impact protein function	Mutation data

measures. *F*-measure is mean of precision and recall in harmonic manner and defined as the following:

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

where the precision is defined as the following:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

and recall is defined as the following:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

In the above equations, TP stands for the number of true positive, FP stands for the number of false positive and FN stands for number of false negative items. We will use *F*-measure as classification quality measure for evaluation of the iMaxDriver.

3. Results

We weighted the modified TRN using the GE data of three cancer types, including breast invasive carcinoma (BRCA), lung squamous cell carcinoma (LUSC) and colon adenocarcinoma (COAD) independently. Then, the list of predicted CDGs was generated using iMaxDriver (Supplementary Datasets S1 and S2). The iMaxDriver^{W} could find 103, 143 and 113 CDGs for BRCA, LUSC and COAD, respectively. Subsequently, the iMaxDriver^{N} can find 88 driver genes in each of BRCA and LUSC tissues and 90 driver genes in COAD. We evaluated our method and the other fifteen methods using cancer gene census (CGC) and functionally validated driver genes provided in Ref. [31] by Kumar et al. and gathered the results for BRCA, LUSC and COAD in Table 3. In each of the tissue types and validation datasets, top three methods with the best prediction results is shown bold. In all of the tissue types, the iMaxDriver^{w} is one of the top three methods with the best results. Moreover, the iMaxDriver^{N} in BRCA and LUSC tissue types is one of the top three methods.

Table 3
The evaluation of the iMaxDriver and the other methods using CGC and Kumar datasets.

Method Name	BRCA				LUSC				COAD			
	Number of predicted drivers		Fraction of predicted drivers		Number of predicted drivers		Fraction of predicted drivers		Number of predicted drivers		Fraction of predicted drivers	
Validation Dataset	Kumar	CGC	Kumar	CGC	Kumar	CGC	Kumar	CGC	Kumar	CGC	Kumar	CGC
iMaxDriver (W)	**27**	**103**	**0.273**	**0.180**	**43**	**143**	**0.434**	**0.252**	**29**	**113**	**0.293**	**0.197**
iMaxDriver (N)	**29**	**88**	**0.293**	**0.153**	**22**	**88**	**0.162**	**0.155**	25	**90**	0.253	**0.157**
ActiveDriver	8	27	0.081	0.047	9	18	0.091	0.031	10	20	0.101	0.034
CoMDP	0	0	0.000	0.000	0	1	0.000	0.001	0	0	0.000	0.000
DawnRank	11	14	0.111	0.024	10	11	0.101	0.019	12	15	0.121	0.026
Dendrix	9	20	0.091	0.034	1	6	0.010	0.010	5	9	0.051	0.015
DriverNet	22	37	0.222	0.064	7	11	0.071	0.019	12	16	0.121	0.027
e-Driver	16	32	0.162	0.055	9	17	0.091	0.030	11	18	0.111	0.031
iPAC	**63**	**250**	**0.636**	**0.437**	**32**	**105**	**0.323**	**0.185**	**63**	**286**	0.636	**0.500**
MDPFinder	5	6	0.051	0.010	0	0	0.000	0.000	1	2	0.010	0.003
MEMO	10	14	0.101	0.024	0	0	0.000	0.000	0	0	0.000	0.000
MSEA	**27**	76	**0.273**	0.132	17	28	0.172	0.049	20	38	0.202	0.066
MutsigCV	11	28	0.111	0.048	5	5	0.051	0.008	8	8	0.081	0.013
NetBox	2	2	0.020	0.003	21	76	0.212	0.134	11	14	0.111	0.024
OncodriveCLUST	2	4	0.020	0.006	11	20	0.111	0.035	4	5	0.040	0.008
OncodriveFM	6	8	0.061	0.013	3	3	0.030	0.005	6	6	0.061	0.010
Simon	8	16	0.081	0.027	3	3	0.030	0.005	**37**	46	**0.374**	0.080

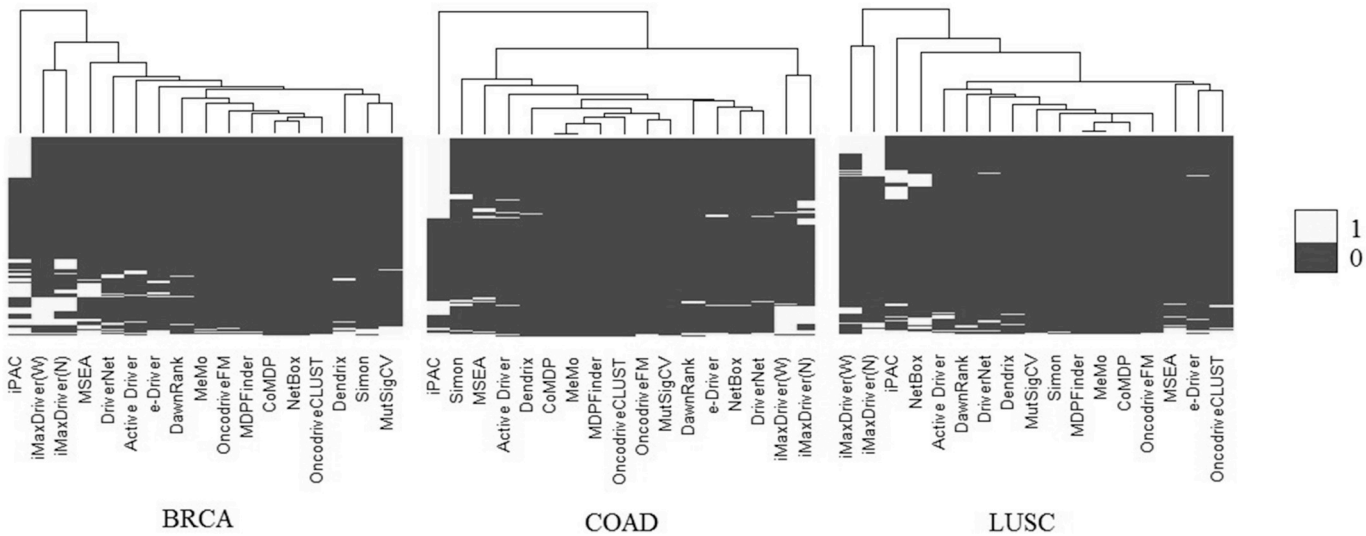

Fig. 4. Binary matrix representation of the genes predicted as CDG by the methods. The bar plot comparing *F*-measure values of the methods is shown for BRCA, LUSC and COAD in Fig. 5.

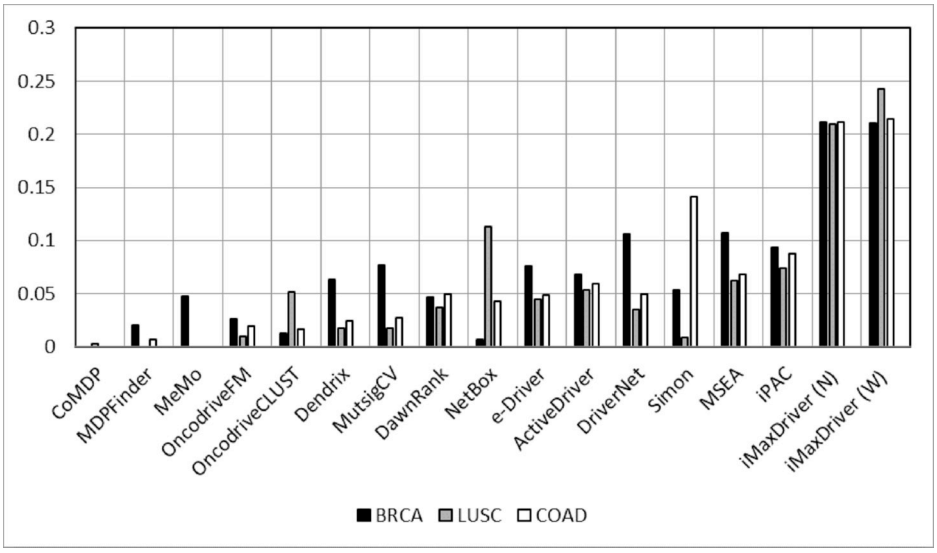

Fig. 5. The *F*-measure of iMaxDriver and other fifteen computational methods proposed for CDG prediction.

Most CDG finding tools predict only a limited number of genes. Although some of these tools predict many CDGs in their output, they are not generally of an acceptable precision. As an example, for BRCA tissue, iPAC and MSEA predict 4821 and 855 genes as CDGs, while their precision is as small as 5.1% and 8.8%, respectively. In contrast, iMaxDriver^{W} predicts 408 genes as CDGs, with a precision value of 33.3%. The binary matrix representation of the genes predicted as CDG by the methods and dendrogram of clustering result are shown for BRCA, LUSC and COAD in Fig. 4.

Furthermore, by comparing the list of classified genes, we can see that more than 38% of the genes classified as driver by iMaxDriver^{W} are reported by none of the other fifteen computational tools. The iMaxDriver^{W} reported 43, 97 and 43 new driver genes in BRCA, LUSC and COAD respectively. Additionally, more than 41% of the genes classified as driver by iMaxDriver^{N} are reported as driver by none of the other fifteen computational tools. Furthermore, the iMaxDriver^{N} reported 41, 68 and 37 new driver genes in BRCA, LUSC and COAD respectively. This means many new driver genes introduced by iMaxDriver. The Venn diagram of the result of gene overlaps among iMaxDriver and union of other fifteen computational tools is shown in Fig. 6. Further, in case of precision and recall iMaxDriver provided acceptable results that is available at Supplementary Figs. S1 and S2.

4. Discussion

The current computational tools available for CDG prediction rely heavily on mutation rates. First group of methods including ActiveDriver [3], CoMDP [12], Dendrix [9], e-Driver [5],

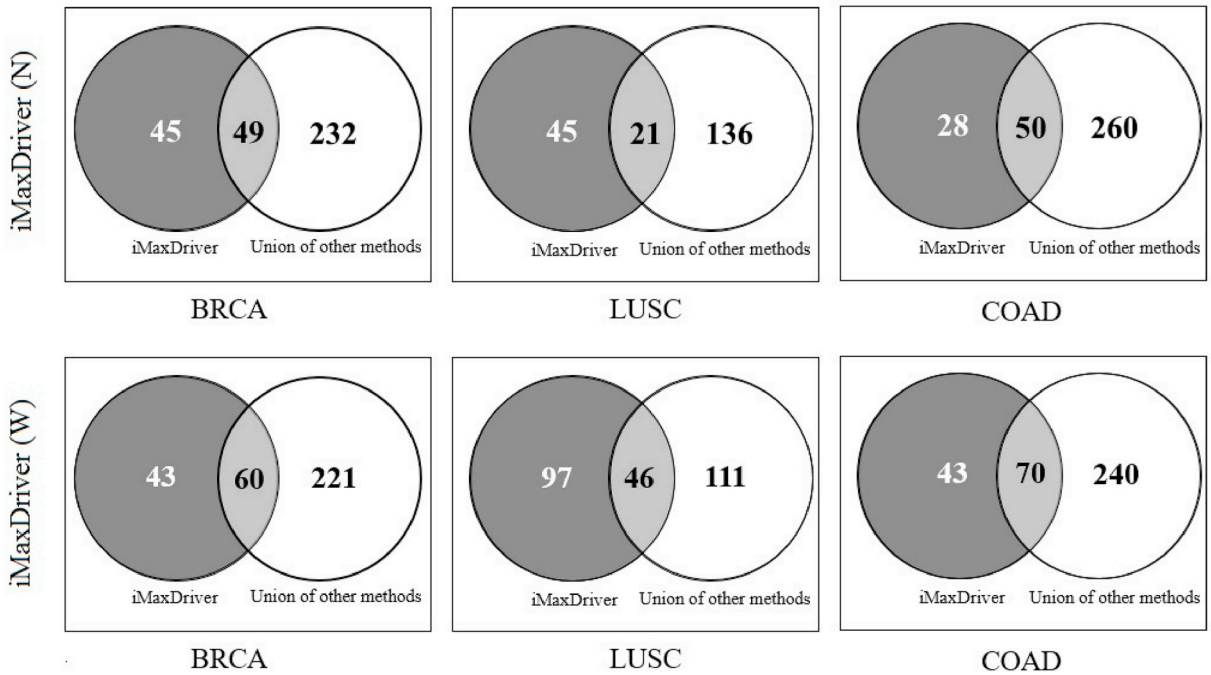

Fig. 6. The Venn diagram for correctly-predicted CDGs using iMaxDriver and the union of correctly predicted CDGs using all other methods.

OncodriveCLUST [8], OncodriverFM [7] and Simon [6] exploit mutation profiles as main feature for CDG prediction. The second group of methods including DawnRank [4], DriverNet [13], iPAC [17], MDPFinder [16], MeMo [10] and NetBox [14] rely on mutation data integrated with other omics data such as GE, network and pathways data etc. MSEA [11] exploits disease-related data in addition to functional genomics and gene network data and finally MutsigCV [15] relies on GE and exome sequence for CDG prediction. In contrast, the iMaxDriver exploits GE data together with TRN data without using mutation data. This strategy is shown to result in better *F*-measure values in comparison with the other state-of-the-art methods. We proposed iMaxDriver^{W} and iMaxDriver^{N} for weighted and non-weighted networks, respectively. The arbitrary values (0.2, 0.5 and 0.8) is used for mapping confidence of relationships for the iMaxDriver^{W}. Changing these values only affects the final results slightly (Supplementary Fig. S3). The random values are used for edge weighting when the network was non-weighted due to lack of information about effect of each transcription factor on the other genes. Furthermore, our results suggest that average of coverage values is a good proxy for the influence of a CDG on the regulation of other genes in iMaxDriver^{N}. The results indicate that iMaxDriver^{W} surpass iMaxDriver^{N} in number of correctly classified CDGs and F-measure since more information has used as input in iMaxDriver^{W} and consequently the results of IM is improved.

The CGC project aims to catalogue genes which causally result in cancer. The results of iMaxDriver method are compared with CGC driver gene list and the precision, recall and *F*-measure of the iMaxDriver is compared with other fifteen computational tools available for CDG prediction. The results indicate that often the recall (and sometimes the precision) of iMaxDriver are one of the best among the tested methods. However, in case of all cancer tissues, the *F*-measure of iMaxDriver is superior compared to all of the other methods (Fig. 4). Furthermore, iMaxDriver shows a relative "orthogonality" to other methods, as a considerable portion of its predicted CDGs are not detected any of the other methods. This indicates that iMaxDriver can find complementary driver genes for state-of-the-art computational CDG prediction tools.

Conflicts of interest

The authors declare that they have no competing interests.

Data availability

Data is presented within the manuscript and the Supplemental Materials.

Software availability

The software is available publicly at https://github.com/majid-rh/iMaxDriver.

Conflicts of interest

None declared.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2019.103362.

Algorithm 1. The iMaxDriver^{W} algorithm.

Input: $G(V, E, W)$: A directed and weighted graph of genes and a list T of threshold value for each gene ID that constructed according to descriptions provided in Subsection 3.1

Output: *Result*: A list of genes sorted by their coverage

```
Result ← an empty set of <GeneID, CoverageCount> with
    initially CoverageCount = 0
AllNodes ←all nodes of G
Calculate MaxIncome as maximum of sum of weights of
    incoming edges to the nodes
for each edge e in E:
    W(e) ← W(e) / MaxIncome
end for
for each node v in V:
    CoverageCount = FindCoverage(v, G, T)
  // finding coverage of v in graph G
    Add <GeneId(v),CoverageCount> to Result
end for
Sort Result by CoverageCount descending
```

Algorithm 2. The iMaxDriver^{N} algorithm.

Input: $G(V, E)$: A non-weighted directed graph of genes and a list T of threshold value for each gene ID

Output: *Result*: A list of genes sorted by their coverage

```
Result ← an empty set of <GeneID, CoverageCount>
with initially CoverageCount = 0
AllNodes ←all nodes of G
for n from 1 to 100:
    for each edge e in E:
        W(e) ← a random value
    end for
    for each node v in V:
        IncomeEdges ← income edges of v
        Sum ← sum of edge weights of IncomeEdges
        for each edge i in IncomeEdges:
            W(i) ← W(i) / Sum
        end for
    end for
    for each node v in V:
        CoverageCount = FindCoverage(v, G, T)
// finding coverage of v in graph G
        Result(v) ← Result(v) + (CoverageCount − Result(v)) / n
    end if
    end for
end for
Sort Result by CoverageCount descending
```

Algorithm 3. iMaxDriver core algorithm (*FindCoverage*).

Input: $G(V, E, W)$ as a directed and weighted graph of genes and list T of threshold values for each gene ID and v as a selected initial node

Output: *ActiveCount*: Number of genes activated by the selected initial node v

```
ActiveNodes ← v
ActiveCount ← 0
While (Count (ActiveNodes) > ActiveCount )
    NewActiveNodes ← empty set
    ActiveCount ← Count(ActiveNodes)
    InactiveNodes ← (AllNodes – ActiveNodes)
    for each node n in InactiveNodes:
        IncomeWeights ← sum of weights of all edges from ActiveNodes to n
        if (IncomeWeights − T(n) ≥ α )
            NewActiveNodes ← NewActiveNodes ∪ n
        end if
    end for
    ActiveNodes ← ActiveNodes ∪ NewActiveNodes
end while
```

References

[1] L. Ding, et al., Somatic mutations affect key pathways in lung adenocarcinoma, Nature 455 (7216) (Oct. 2008) 1069–1075.
[2] L. Mularoni, R. Sabarinathan, J. Deu-Pons, A. Gonzalez-Perez, N. López-Bigas, OncodriveFML: a general framework to identify coding and non-coding regions with cancer driver mutations, Genome Biol. 17 (1) (2016) 128.
[3] J. Reimand, O. Wagih, G.D. Bader, The mutational landscape of phosphorylation signaling in cancer, Sci. Rep. 3 (2013) 2651.
[4] J.P. Hou, J. Ma, DawnRank: discovering personalized driver genes in cancer, Genome Med. 6 (7) (2014) 56.
[5] E. Porta-Pardo, A. Godzik, e-Driver: a novel method to identify protein regions driving cancer, Bioinformatics 30 (21) (2014) 3109–3114.
[6] A. Youn, R. Simon, Identifying cancer driver genes in tumor genome sequencing studies, Bioinformatics 27 (2) (2010) 175–181.
[7] A. Gonzalez-Perez, N. Lopez-Bigas, Functional impact bias reveals cancer drivers, Nucleic Acids Res. 40 (21) (2012) e169–e169.
[8] D. Tamborero, A. Gonzalez-Perez, N. Lopez-Bigas, OncodriveCLUST: exploiting the positional clustering of somatic mutations to identify cancer genes, Bioinformatics 29 (18) (2013) 2238–2244.
[9] F. Vandin, E. Upfal, B.J. Raphael, De novo discovery of mutated driver pathways in cancer, Genome Res. 22 (2) (Feb. 2012) 375–385.
[10] G. Ciriello, E. Cerami, C. Sander, N. Schultz, Mutual exclusivity analysis identifies oncogenic network modules, Genome Res. 22 (2) (Feb. 2012) 398–406.
[11] D. Arneson, A. Bhattacharya, L. Shu, V.-P. Mäkinen, X. Yang, Mergeomics: a web server for identifying pathological pathways, networks, and key regulators via multidimensional data integration, BMC Genomics 17 (1) (2016) 722.
[12] J. Zhang, L.-Y. Wu, X.-S. Zhang, S. Zhang, Discovery of co-occurring driver pathways in cancer, BMC Bioinf. 15 (1) (2014) 271.
[13] A. Bashashati, et al., DriverNet: uncovering the impact of somatic driver mutations on transcriptional networks in cancer, Genome Biol. 13 (12) (2012) R124.
[14] E. Cerami, E. Demir, N. Schultz, B.S. Taylor, C. Sander, Automated network analysis identifies core pathways in glioblastoma, PLoS One 5 (2) (2010) e8918.
[15] M.S. Lawrence, et al., Mutational heterogeneity in cancer and the search for new cancer-associated genes, Nature 499 (7457) (2013) 214.
[16] J. Zhao, S. Zhang, L.-Y. Wu, X.-S. Zhang, Efficient methods for identifying mutated driver pathways in cancer, Bioinformatics 28 (22) (2012) 2940–2947.
[17] M.R. Aure, et al., Identifying in-trans process associated genes in breast cancer by integrated analysis of copy number and expression data, PLoS One 8 (1) (2013) e53014.
[18] J.M. Vaquerizas, S.K. Kummerfeld, S.A. Teichmann, N.M. Luscombe, A census of human transcription factors: function, expression and evolution, Nat. Rev. Genet. 10 (4) (2009) 252.
[19] Y. Li, J. Fan, Y. Wang, K. Tan, Influence maximization on social graphs: a survey, IEEE Trans. Knowl. Data Eng. 30 (10) (Oct. 2018) 1852–1872.
[20] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.
[21] Z.-P. Liu, C. Wu, H. Miao, H. Wu, RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse, Database 2015 (Jan. 2015).
[22] H. Han, et al., TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions, Nucleic Acids Res. 46 (D1) (Jan. 2018) D380–D386.
[23] L. Gautier, L. Cope, B.M. Bolstad, R.A. Irizarry, "affy—analysis of Affymetrix GeneChip data at the probe level, Bioinformatics 20 (3) (Feb. 2004) 307–315.
[24] I.-F. Chung, et al., DriverDBv2: a database for human cancer driver gene research, Nucleic Acids Res. 44 (D1) (2015) D975–D979.
[25] C.G.A. Network, Comprehensive molecular portraits of human breast tumours, Nature 490 (7418) (2012) 61.
[26] C.G.A.R. Network, Comprehensive genomic characterization of squamous cell lung cancers, Nature 489 (7417) (2012) 519.
[27] C.G.A. Network, Comprehensive molecular characterization of human colon and rectal cancer, Nature 487 (7407) (2012) 330.
[28] J.N. Weinstein, et al., The cancer genome atlas pan-cancer analysis project, Nat. Genet. 45 (10) (2013) 1113.
[29] P.A. Futreal, et al., A census of human cancer genes, Nat. Rev. Cancer 4 (3) (2004) 177.
[30] X. Robin, et al., pROC: an open-source package for R and S+ to analyze and compare ROC curves, BMC Bioinf. 12 (1) (2011) 77.
[31] R.D. Kumar, A.C. Searleman, S.J. Swamidass, O.L. Griffith, R. Bose, Statistically identifying tumor suppressors and oncogenes from pan-cancer genome-sequencing data, Bioinformatics 31 (22) (Nov. 2015) 3561–3568.