Supplementary Information:

KG4SL: Knowledge Graph Neural Network for Synthetic Lethality Prediction in Human Cancers

Shike Wang [1,†], Fan Xu [1,†], Yunyang Li [2], Jie Wang [1], Ke Zhang [1,3], Yong Liu [4], Min Wu [5,*] and Jie Zheng [1, 6,*]

[1]School of Information Science and Technology, ShanghaiTech University, Shanghai, 201210, China,

[2]School of Life Science and Technology, ShanghaiTech University, Shanghai, 201210, China,

[3]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, 200050, China,

[4]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, 639798, Singapore,

[5]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 138632, Singapore,

[6]Shanghai Engineering Research Center of Intelligent Vision and Imaging, Shanghai, 201210, China.

[*]Corresponding e-mail: wumin@i2r.a-star.edu.sg; zhengjie@shanghaitech.edu.cn.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint Authors.

Exp 1: In KG4SL, all genes in the SL matrix obtain their representations through the well-designed knowledge graph, which is crucial to SL pair prediction. In Section 3.3, we investigated the importance of SynLethKG for KG4SL. Here, we further conducted experiments to evaluate the performance of KG4SL using different data sources of protein-protein interaction (PPI) relationships.

The Gene-interacts-Gene (GiG) relationships are PPI data collected from STRING database. Then, we replaced these PPI relationships from STRING with those from the MINT database. To this end, we first removed all the GiG edges in our original knowledge graph SynLethKG. We then collected PPI data of Homo Sapiens from MINT and obtained the 59,910 PPIs. We mapped these PPIs to SynLethKG and only kept PPIs between proteins that are already in SynLethKG. Finally, 30,421 PPIs from MINT were added into SynLethKG.

Table R1 summarizes the performance of KG4SL based on different sources of PPI data. We can observe that the performance of KG4SL has only a slight decline. This might be caused by 79.5% reduction in the number of PPI relationships (i.e., from 148,379 to 30,421). In addition, these experimental results also demonstrate that our knowledge graph has been carefully constructed to depict all kinds of relations very well.

Table R1: The performances of KG4SL based on different sources of PPI data.

	AUC	AUPR	F1
KG4SL (STRING)	**0.9470**±0.0003	**0.9564**±0.0005	**0.8877**±0.0017
KG4SL (MINT)	0.9410±0.0006	0.9525±0.0005	0.8801±0.0007

Exp 2: By default, when building the KG4SL model, we treated all types of relationships in SyLethKG as bidirectional (denoted by KG4SL-Bi). Through semantic analysis, we noted that only the relationships of types Gene-interacts-Gene and Gene-covaries-Gene are bidirectional and the remaining 22 relationship types in SyLethKG are monodirectional. To study the impact of edge directionality on the performance of KG4SL, we re-trained the KG4SL model by treating the Gene-interacts-Gene and Gene-covaries-Gene relationships as bidirectional and all other 22 relationships as monodirectional (denoted by KG4SL-Mono). Table R2 summarizes the performance of KG4SL under different settings of edge directionality. Compared with the result of bidirectional relationships, the AUC and AUPR of monodirectional relationships are only improved by 0.01 %, while the F1 decreased by 0.13 %. Thus, we are more inclined to use bidirectional relationships to build the KG4SL.

Table R2: The performance of KG4SL under different settings of edge directionality.

	AUC	AUPR	F1
KG4SL-Bi	0.9470±0.0003	0.9564±0.0005	**0.8877**±0.0017
KG4SL-Mono	**0.9471**±0.0006	**0.9565**±0.0005	0.8857±0.0018

Exp 3: To the best of our knowledge, there is no existing work estimating the ratio between positive and negative SL pairs. Thus, following previous study (Huang et al., 2020), we empirically set the ratio between positive and negative SL pairs as 1:1 in this work.

Nonetheless, according to your kind advice, we conducted additional experiments on various imbalance sample ratios to simulate real situations, and compared our model with three baseline models (due to the limited time, we are not able to run all the baselines). The ratios of positive samples to negative samples range from 1:1, 1:2, 1:5 to 1:10 on both ***the training and test sets***. As shown in Table R3, the more unbalanced are the training data, the worse is the performance of the models. The tendency is common in machine learning algorithms, as models have poor predictive performance on the minority class. On the other hand, KG4SL has the best or the second best performance under different sample ratios, while GraphSAGE performs better than KG4SL in terms of AUPR and F1 when the ratio is 1:10. Overall, the results in Table R3 indicate that KG4SL would also work well on a real SL data distribution.

References

Huang, Y.-a. et al. (2020). Graph convolution for predicting associations between miRNA and drug resistance. Bioinformatics. 36(3), 851-858.

Table R3: The effect of the unbalanced training data on the results

		AUC	AUPR	F1
1:1	KG4SL	**0.9470**±0.0003	**0.9564**±0.0005	**0.8877**±0.0017
	DeepWalk	0.8391±0.0016	0.8557±0.0017	0.7544±0.0028
	GraphSAGE	0.8398±0.0291	0.8775±0.0188	0.8569±0.0236
	GCN	0.8329±0.0172	0.8727±0.0110	0.8508±0.0136
1:2	KG4SL	**0.9442**±0.0007	**0.9278**±0.0011	**0.8552**±0.0018
	DeepWalk	0.8338±0.0022	0.7712±0.0025	0.6861±0.0016
	GraphSAGE	0.8215±0.0129	0.8669±0.0074	0.8454±0.0094
	GCN	0.7278±0.1172	0.8282±0.0415	0.7892±0.0640
1:5	KG4SL	**0.9428**±0.0015	**0.8677**±0.0022	0.8069±0.0058
	DeepWalk	0.8404±0.0020	0.6479±0.0084	0.6088±0.0049
	GraphSAGE	0.7871±0.0213	0.8500±0.0108	**0.8234**±0.0147
	GCN	0.6833±0.0907	0.8095±0.0352	0.7625±0.0529
1:10	KG4SL	**0.9409**±0.0010	0.8095±0.0016	0.7373±0.0087
	DeepWalk	0.8390±0.0012	0.5357±0.0064	0.5356±0.0046
	GraphSAGE	0.7530±0.0250	**0.8345**±0.0117	**0.8016**±0.0165
	GCN	0.6670±0.0770	0.8015±0.0270	0.7515±0.0424

Exp 4: Indeed, a method for data splitting could make a big difference to the results. The gene leave-out strategy you mentioned above is exactly more in line with the biological scenario, where many genes do not have any known SL partners. According to your suggestion, we conducted another cross validation experiment based on the gene leave-out strategy as you mentioned above. First, all genes in the SL matrix were randomly split into five parts, and we took one part as test genes each time. Then, the pairs that contain at least one test gene are grouped into the test set, and otherwise as training pairs. Apart from applying this method on KG4SL, we also compared its performance with some other baselines. As shown in Table R4, our method performs better than baseline methods in terms of AUC and is comparable to GraphSAGE on AUPR and F1 metrics, demonstrating that KG4SL is also effective for the more challenging gene leave-out data splitting scenario.

Table R4: Gene leave-out prediction results

	AUC	AUPR	F1
KG4SL	**0.7090**±0.0246	0.7437±0.0257	0.6686±0.0311
GRSMF	0.6164±0.0293	0.6112±0.0221	0.6068±0.0249
GraphSAGE	0.5538±0.0502	**0.7581**±0.0009	**0.6854**±0.0189

Exp 5: KRAS is one of the most commonly mutated oncogene in human cancer. Unfortunately, it is still challenging to develop small molecule inhibitors that directly act on KRAS (Aguirre et al., 2018). Synthetic lethality based treatment is a promising approach for KRAS-mutated cancer (Pang et al., 2017). Thus, in our case study, we focused on SL pairs containing KRAS as the primary gene. In the prediction results of KG4SL, we observed that among the top 1,000 SL pairs, 29 SL pairs containing KRAS have been reported in the literature. As shown in Table R5, these pairs have been validated in different ways, e.g. row3 (KRAS-EZH2) is validated by RNAi screening (Wang et al., 2014), and row11 (KRAS-CMPK1) is validated by CRISPR screening (Martin et al., 2017). Most of them are validated based on large-scale wet-lab on specific cell lines. It is worth mentioning that some pairs are validated in small-scale in vitro, or even in vivo. Taking one SL pair (KRAS and DDR1) as an example, Jeitany et al. (Jeitany et al., 2018) showed that DDR1 depletion strongly inhibited the invasive capacities of KRAS-mutated HCT116 in vitro, and Nokin et al. (Nokin et al., 2020) found that combining DDR1 inhibition with chemotherapy enhanced cell death of KRAS-mutant lung tumors in vivo. KRAS is involved in the transmission of cellular signals. Meanwhile, DDR1 plays a key role in the communication of cells with their microenvironment. Therefore, the similarity of their functions makes them reasonable for SL pairs. In addition, in order to find the characteristics of the results predicted by KG4SL, we also performed enrichment analysis on these 29 KRAS partner genes. According to the enrichment analysis, we found that there are 22 genes related to protein binding function, and 7 genes involved in signal transduction. This indicates that our model had better consider the signaling pathways when predicting SL pairs.

We have included this case stuy on KRAS-related SL pairs in our supplementary materials.

References

Aguirre, A. J. et al. (2018). Synthetic lethal vulnerabilities in KRAS-mutant cancers. Cold Spring Harbor perspectives in medicine, 8(8), a031518.

Pang, X. et al. (2017). Defeat mutant KRAS with synthetic lethality. Small GTPases, 8(4), 212-219.

Wang, X. et al. (2014). Widespread genetic epistasis among cancer genes. Nature communications, 5, 4828.

Martin, T. D. et al. (2017). A Role for Mitochondrial Translation in Promotion of Viability in K-Ras Mutant Cells. Cell reports, 20(2), 427–438.

Jeitany, M. et al. (2018). Inhibition of DDR 1-BCR signalling by nilotinib as a new therapeutic strategy for metastatic colorectal cancer. EMBO molecular medicine, 10(4), e7918.

Nokin, M. J. et (2020). Inhibition of DDR1 enhances in vivo chemosensitivity in KRAS-mutant lung adenocarcinoma. JCI insight, 5(15).

Table R5: Top predicted SL pairs containing KRAS with literature support.

Number	Rank	Gene A	Gene B	Pubmed
1	3	KRAS	DDR1	24104479
2	17	KRAS	BCR	27655641
3	74	KRAS	EZH2	25407795
4	89	KRAS	SSH3	24104479
5	96	KRAS	CMPK1	24104479
6	168	KRAS	DLGAP5	24104479
7	189	KRAS	MAP3K11	27655641
8	216	KRAS	CDH1	27655641
9	236	KRAS	BIRC3	25407795
10	275	KRAS	DLG1	24104479
11	286	KRAS	GGA3	28700943
12	299	KRAS	RIN2	24104479
13	310	KRAS	FH	25407795
14	323	KRAS	PSMC3	28700943
15	364	KRAS	ALPK1	27655641
16	483	KRAS	MMADHC	24104479
17	522	KRAS	CREBZF	22613949
18	586	KRAS	CPSF1	24104479
19	706	KRAS	NFYB	28700943
20	707	KRAS	MAP2K4	25407795
21	818	KRAS	DHFR	28700943
22	830	KRAS	RBBP8	24104479
23	839	KRAS	CYP1B1	22613949
24	860	KRAS	MAP2K7	24104479
25	887	KRAS	DDX51	28700943
26	908	KRAS	SRC	27655641
27	913	KRAS	CFLAR	24104479
28	932	KRAS	SMC3	28700943
29	939	KRAS	BCLAF1	28700943

Exp 6: We divided the SL pairs in the test set into two groups according to the degree of each node computed on the training set. In one group, the degrees of both nodes in one SL pair are lower than 15, and the rest belong to the other group. We set the threshold to 15 because we observed from the data that all SL pairs can be classified into two groups of balanced sizes only when the threshold was set to 15. Table R6 shows the performance of our model for the SL pairs in the two groups. From the table, we can see that the performance of KG4SL on the SL pairs that involve genes with fewer known SL partners in the training set is poorer than the performance on the rest SL pairs. It suggests that SL pairs in sparse parts of the SL network would be more difficult to detect.

Table R6: The effect of the node degrees on the results.

Degree	AUC	AUPR	F1
<15	0.7938	0.5385	0.4912
Otherwise	0.9435	0.9896	0.9616